[illegible] Support Around The World

The Computer Journal

Issue Number 65 January/February 1994 US$4.00

Small System Support

European Beat

DR S-100

Real Computing

Support Groups

PC/XT Corner

Little Circuits

Levels of Forth

Sinclair ZX-81

The Computer Corner

ISSN # 0748-9331

TCJ - For Having Fun With Any Computer!

68XXX COMPUTER PRODUCTS From Peripheral Technology

68000 System Boards with 4 Serial/ 2 Parallel Ports, FDC, and RTC.

PT68K4-16 with 1MB	$299.00
PT68K2-10 w/ 1MB (Used)	$149.00

REX Operating System Included

OS9 V2.4 Operating System $299.00
With C, Editor, Assembler/Linker
SCULPTOR V1.14:6 for Business Software Development - requires any version of OS9/68K. $79.00

<u>Other 68XXX products available!</u>

1480 Terrell Mill Rd. #870
Marietta, GA 30067
404/973-2156

Cross-Assemblers as low as $50.00
Simulators as low as $100.00
Cross-Disassemblers as low as $100.00
Developer Packages

as low as $200.00(a $50.00 Savings)

A New Project

Our line of macro Cross-assemblers are easy to use and full featured, including conditional assembly and unlimited include files.

Get It To Market--FAST

Don't wait until the hardware is finished to debug your software. Our Simulators can test your program logic before the hardware is built.

No Source!

A minor glitch has shown up in the firmware, and you can't find the original source program. Our line of disassemblers can help you re-create the original assembly language source.

Set To Go

Buy our developer package and the next time your boss says "Get to work.", you'll be ready for anything.

Quality Solutions

PseudoCorp has been providing quality solutions for microprocessor problems since 1985.

BROAD RANGE OF SUPPORT

- Currently we support the following microprocessor families (with more in development):

Intel 8048	RCA 1802,05	Intel 8051	Intel 8096
Motorola 6800	Motorola 6801	Motorola 68HC11	Motorola 6805
Hitachi 6301	Motorola 6809	MOS Tech 6502	WDC 65C02
Rockwell 65C02	Intel 8080,85	Zilog Z80	NSC 800
Hitachi HD64180	Motorola 68000,8	Motorola 68010	Intel 80C196

- All products require an IBM PC or compatible.

So What Are You Waiting For? Call us:

PseudoCorp

Professional Development Products Group
716 Thimble Shoals Blvd, Suite E
Newport News, VA 23606
(804) 873-1947 FAX: (804)873-2154

Journey with us to discover the shortest path between programming problems and efficient solutions.

The Forth programming language is a model of simplicity: In about 16K, it can offer a complete development system in terms of compiler, editor, and assembler, as well as an interpretive mode to enhance debugging, profiling, and tracing.

As an "open" language, Forth lets you build new control-flow structures, and other compiler-oriented extensions that closed languages do not.

Forth Dimensions is the magazine to help you along this journey. It is one of the benefits you receive as a member of the non-profit Forth Interest Group (FIG). Local chapters, the GEnie™ Forth RoundTable, and annual FORML conferences are also supported by FIG. To receive a mail-order catalog of Forth literature and disks, call 510-89-FORTH or write to:
Forth Interest Group, P.O. Box 2154, Oakland, CA 94621.
Membership dues begin at $40 for the U.S.A. and Canada.
Student rates begin at $18 (with valid student I.D.).

GEnie is a trademark of General Electric.

SAGE MICROSYSTEMS EAST

Selling and Supporting the Best in 8-Bit Software

Z3PLUS or NZCOM (now only $20 each)
ZSDOS/ZDDOS date stamping BDOS ($30)
ZCPR34 source code ($15)
BackGrounder-ii ($20)
ZMATE text editor ($20)
BDS C for Z-system (only $30)
DSD: Dynamic Screen Debugger ($50)
4DOS "zsystem" for MSDOS ($65)
ZMAC macro-assembler ($45 with printed manual)

Kaypro DSDD and MSDOS 360K FORMATS ONLY

Order by phone, mail, or modem and use Check, VISA, or MasterCard. Please include $3.00 Shipping and Handling for each order.

Sage Microsystems East
1435 Centre Street
Newton Centre MA 02159-2469
(617) 965-3552 (voice 7PM to 11PM)
(617) 965-7259 (pw=DDT)
(MABOS on PC-Pursuit)

The Computer Journal

Founder
Art Carlson

Editor/Publisher
Bill D. Kibler

Technical Consultant
Chris McEwen

Contributing Editors
Herb Johnson
Charles Stafford
Brad Rodriguez
Ronald W. Anderson
Helmut Jungkunz
Tim McDonough
Frank Sergeant
JW Weaver
Richard Rodman
Jay Sage
Tilmann Reh

The Computer Journal is published six times a year and mailed from *The Computer Journal*, P. O. Box 535, Lincoln, CA 95648, (916) 645-1670.

Opinions expressed in *The Computer Journal* are those of the respective authors and do not necessarily reflect those of the editorial staff or publisher.

Subscription rates within the US: $24 one year (6 issues), $44 two years (12 issues). All funds must be in U.S. dollars drawn on a U.S. bank. Send subscription, renewals, address changes, or advertising inquires to: *The Computer Journal*, P.O. Box 535, Lincoln,CA 95648.

Registered Trademarks

TCJ The Computer Journal

Issue Number 65 January/February 1994

EDITOR'S COMMENTS

Welcome to the eleventh year of ***The Computer Journal.*** This issue has some new items for your interest. The major addition is TWO new regular columns. We start off our regular features with a new writer to TCJ, Helmut Jungkunz. I have called his column "The European Beat." Helmut's assignment is to report on Z80 and CP/M happenings in Europe. Actually I don't expect him to limit his discussion to only those two topics, but then there seems to be a lot still happening in Europe that is not PC based. Perhaps Helmut can help explain why the PC is so hated over there.

Speaking of PC's is our next new regular column. One of TCJ's favorite writers is back in splendid form as Frank Sergeant starts his support column for using PC and XT systems. This is not your normal support column as you will find out when you read the first of many articles on mastering the mysterious and dreaded PC's architecture.

Since this was the holiday season, several of our writers took the time off to be with relatives and friends. One who didn't take the time off was our still new Ronald Anderson. This time Ron fills us in on more history and specially that relating to serial communications on the older 6800/09 systems. Ron also comments on a few current topics and don't forget that serial communications is the same now as it was eleven years ago!

AS I promised, this is a special issue on the ZX-81. When you finish understanding serial history, checkout the comments, books, and emulators for an amazing Z80 product. After reading and preparing this article (thanks to the many who contributed your letters and schematics), I have some new ideas about what makes a good learning hardware platform. The Centerfold this time is the supporting schematics for the Sinclair ZX-81.

After you work your way through Z80 history, Rick Rodman gives us a quick report on some major projects he is working on. It is starting to look like we may have a multi-system network in TCJ's future thanks to RIck.

For you hardware junkies, Dave Baldwin has started the first in a series of "Little Circuit" tutorials. The first one talks about the often overlooked, but very important, reset circuit. Dave explains why some flaky boards just need a better reset circuit to solve their problems. Dave also has two short letters in Reader to Reader that you shouldn't miss as well. Thanks for the article Dave!

We catch up next with JW Weaver as he gives us some changes suggested by Tilmann Reh in "Support Groups for the Classics." JW also shows us some hardware "how-to" using his Kaypro and it's expansion connector.

While at Forth Day 1993, I talked to Glen Haydon about learning Forth. Glen had written some handy advice for would be Forth people which is valid for any one interested in programming languages. You will find Glen's excellent words just before DR. S-100.

Finding himself almost last is Herb Johnson as Dr. S-100 (sorry Herb, but you lost out on the coin toss as I was placing articles.) Herb took this issue to catch up on some letters and their questions. Herb's article is a must read if you get your systems on the used market.

The real tail end of the issue, is of course my own column, "The Computer Corner." This time around I comment on OS/2, a new system, and mainly my new job.

No More Disks!

That is right, TCJ will no longer sell *Micro Cornucopia* Kaypro Disks. My agreement with Dave Thompson was doing just find when I had more time. With the magazine growing, that time keeps getting less for items like older disks.

Have no fear however, as David McGlone of Lambda Software Publishing and *The Z-Letter* has taken them over. When David moved from the San Francisco to Eugene, and indicated a willingness to help TCJ with production problems, it got me thinking to link him with Dave (also living in Eugene). The end results of the effort is more than just off loading some disk copying. David now has the rights to all the old *MicroC* back issues as well. So if you need some article lost in one of the old *MicroC's* give David a call, his prices are very good. (Ad on page 35.)

A New Advertiser!

Welcome a new TCJ supporting advertiser, Parallax. They start with this issue telling us about their products. You will find their first ad on the back cover. I think this is their current hot seller, a BASIC single board computer for $39. Now for all those who though BASIC was dead, check them out.

Hopefully our new advertising rates will get more money coming in. This issue could have been about ten pages longer with Reader to Reader letters. People who say and think that supporting older systems is a dead end project, should look at my mail sometime. I get so many letters (can't find my desk due to them, honestly it is the letters - not me) thanking me for still providing support when all others have given up, (both readers and publishers) on the older and classic systems.

Well stay tuned, as we have plenty in this issue to keep you busy reading.

READER to READER

Letters to the Editor

All Readers

MINI Articles

Bradford J. Rodriguez

is

Deliriously Gratified to announce that he has Passed (with distinction) his Comprehensive Examination.

McMaster University, December 9th, 1993.

Normal life will now resume.

Dear Bill,

David Klink's letter in TCJ #64 makes this news especially valuable to TCJ readers: I recently received JDR Microdevices' latest catalog. Or I should say *catalogs*, because they've started a full catalog just for components! We're talking 60 pages of the stuff TCJ readers need to build their projects, and even *I* was impressed with their selection of ICs. And, bless 'em, they still have NO minimum order, and they eagerly solicit international business, with toll-free fax numbers for a few dozen countries! Write 'em and ask for the "Components Edition" -- JDR Microdevices, 2233 Samaritan Drive, San Jose, CA 95124-4405, order phone (800) 538-5000, (408) 559-1200 outside the U.S., U.S. fax (800) 538-5005.

I also got the latest flyer from B.G.Micro, and it has lots of new goodies (including 2681s for $3, if you're building the Scroungemaster I.) For some reason they can't ship to Mexico or Puerto Rico, but other countries seem ok; $10 minimum on credit card orders. B.G.Micro, P.O. Box 280298, Dallas TX 75228, (214) 271-5546, fax (214) 271-2462.

For everything *other* than ICs, don't forget Digi-Key, 701 Brooks Ave. South, P.O. Box 677, Thief River Falls, MN 56701-0677, phone (800) 344-4539 (even from Canada), fax (218) 681-3380, no minimum order but a service charge on small orders. (They do sell ICs, but their microprocessor selection is poor and their prices are usually higher.)

Alas, Jameco seems to be continuing their trend away from components and small orders (minimum order now $30). But get on their mailing list, because when they close out parts they offer some terrific deals! Jameco, 1355 Shoreway Road, Belmont, CA 94002, order phone (800) 831-4242, order fax (800) 237-6948, customer service/international (415) 592-8097.

Re. the 6809 project, I have just acquired a pile of 68B09s that I'll sell to TCJ readers for US$2.25 or Cdn$3.00 each. This is the 2 MHz part, twice the speed of the "vanilla" 6809. The price includes postage to the US or Canada; please send US or Canadian checks or money orders only. (Residents of other countries please contact me for price.)

I grew up out in the sticks and I live in a relative boondocks now, so I try to design my TCJ projects *only* with parts available mail-order (and preferably from more than one supplier). I blush to admit that I'm not familiar with the 68306 David mentions. But I'm a bona fide consultant, and even *I* have trouble prying samples from manufacturers. I've also seen too many announced products never appear, or appear briefly and then be discontinued. So for TCJ I won't use any part less than, say, five years old...and I'll try always to name suppliers.

O.k. Bill, here's my contribution to the language wars:

Forth is of course my preference. It's widely available, easy to port, efficient on small machines, and intermixes easily with assembler. Not only is the language easy to understand (once you break a few habits), but the language *translator* is easy to understand. I can explain a Forth compiler to a neophyte in a few hours. I don't think I could explain a C compiler in less than a university course. Plus, it's easy to extend Forth into object- oriented programming, symbolic processing, or whatever TCJ readers may need to learn next.

My second choice is Small C. C has always been my second favorite language, and IMHO (In My Humble Opinion) Ron Cain's original Small C article brought C to the masses. A C compiler small enough to comprehend! Marvelous! Considering Carey Bloodworth's excellent commentary, perhaps a "new" Small C; maybe producing "F-code" instead of P-code? (Oops, the name "F-code" is already being used.) For small & slow systems, the Intermediate Language could be interpreted, instead of compiled to machine code.

I disagree with Tilmann Reh about the clarity of Pascal or its derivatives. It may be clear for simple algorithms, but as soon as pointers get involved it becomes fabulously cryptic. To this day, I struggle with the balanced- tree routines in Wirth's *Algorithms and Data Structures*, although Wirth's text and illustrations are quite clear. (For the representation of algorithms in Forth, may I recommend *Scientific Forth* by my good friend Dr. Julian Noble? Available from the Forth Interest Group, of course.)

Whatever you choose, make sure the language supports direct access to memory and I/O locations. It has to be usable with small (classic or embedded) computers -- thus forbidding GNU C -- and the source code should be freely available (public domain). For these reasons I suspect Forth and Small C are your only real alternatives.

Regards, Brad

Congratulations Brad on passing your oral exams. Looks like your on the home stretch toward that Doctorate!

On finding parts, I have heard from several people. The most interesting comment was, that major wholesalers have started taking orders from anybody with a credit card. Add the people you listed and I think the availability of parts is far better than I thought. My guess is the competition has gotten very steep with fewer sales, so all vendors are trying to get those few buyers anyway they can.

The 68306 is suppose to be a complete CPU. All you need to add is memory chips, as all the glue chips are built in. Should make 3 and 4 chip system possible. I had heard that all production was being consumed and none were for sale, but several vendors say they have some available for under $10. All that is listed elsewhere.

I keep reminding people, that learning on 5 year old technology is easier and just as valid as learning on todays just made parts. As simple as the 68306 is suppose to be, our readers should be able to port everything you have done to it over a single weekend. Although I would prefer using the 6809 version simply because I have some sitting unused in my parts box.

I think that saying Forth is my first choice is probably getting redundant. I agree about Pascal being more difficult than Forth once you get past the ''HUMP.'' One book that helped me get over the hump was ''Thinking Forth'' by Leo Brodie. This book is helpful for any programmer of any language, but it helps would be Forth learners to see a bit of the ''thinking process'' involved. Forth is more than a just language to be mastered, it is also a simplified thinking process. When people get over the hump, they usually exclaim that they expected it to be as complex as C, and once they learned that it was not, they found it a very simple language to master.

I started looking at Small-C just for the reasons you outlined (small and learnable). The main concern is, unlike Forth's power in a small package, a small package of C usually doesn't give you enough horsepower to get the job done. As you say, maybe a home grown version with small size and big horsepower is possible. I keep watching my mail in hopes that someone will send me just that item.

Keep up the good work! Bill Kibler.

Bill --

Merry Christmas! I just received Issue #63 of TCJ and it looks better than ever. After many months of being in ''crunch mode'' at my new job things are settling down a bit and I've been thinking about some of the projects I'd wanted to work on for TCJ.

The first is real down and dirty, good for beginners although it is for the PC. I recently worked on a real simple control system that used the handshaking lines on a standard serial port to read two digital inputs and control a triac and a relay for some output switching. The software and hardware are both simple but useful.

The second project is the small serially interfaced data acquisition system I talked about before. I have the parts collected and am just starting to build hardware and design software. I can probably send you the first article for #66. The second one would follow about a month behind if you're interested.

How are things going with TCJ? I hope that it's becoming more than just a ''labor of love'' even if you're not getting rich. Hope you have a nice Holiday season, Tim McDonough

Bill --

I've been reading the discussions about choosing a common language for various TCJ projects. Looking over the various comments it seems pretty clear that while trying to have a ''standard'' is a noble idea, it may not be workable.

Personally, I use C and C++ for many projects. I also use assembler and PLM-51 for a great many. Part of the attraction of a publication like TCJ is getting a look at ideas and implementations that are not in the C Language/MSDOS/PC Architecture mainstream of small computers. I like the variety. I even like (trying) to sift through the FORTH articles! There's nothing like a different perspective to get the creative juices going.

The mention of Dave Dunfield's products seemed a little bit outdated. Dave (Dunfield Development Systems) has versions of his MICRO-C Development system for many processors including the 6809, 8080/Z80, 68HC11, 8086, 8051, etc. Version 3.2, due to be released around January/February of 1994 offers more features than ever including support for some 32-bit math even on the 8-bit processors. For those CPUS Dave doesn't support or for those who want to host MICRO-C on an operating system other than MSDOS, he also sells a ''porting package'' that takes a lot of the work out of making a new code generator for your favorite microprocessor. Dave can be reached via the Internet as ddunfield@bix.com.

Tim McDonough. Internet:
tmcdonough@bix.com

Thanks Tim for not falling completely away from TCJ, your fans have been looking for your articles. The small projects are just the things we need to help those learn about device to device communications.

I hope that many of our readers still send in comments on the Small-C topic, as I felt the bottom line was not very hopeful. Your mention of Dunfield sounds like you converse regularly. How about asking him for an article (or you) reviewing his product line, maybe from a ''why this is better than Small-C.'' I have also had comments that some would *rather just do it in Forth than deal with Small-C's problems.*

I am still open to Small-C, but feel a very important aspect is the operating system it all sits on. One concept of finding a language was also coming up with a universal operating system that could be ported to all 8 bit systems using the same language. Does the Dunfield system do that?

Thanks again and we are all looking forward to your articles. Bill Kibler.

Hello Bill,

meanwhile I browsed through the new TCJ #64. I have some comments.

First, I have some general comments about your work as the editor of TCJ. There are some details (concerning article contents and typesetting) that still can be improved:

a) We want TCJ to be readable for computer newcomers as well as old wizards. This should also apply to americans and non-americans. Please take care to not simply use abbreviations which are known only in the USA! What are degrees like B.S., B.A. and G.R.E.? What, for heavens sake, is a gizmo? You may print this comment in the Reader-To-Reader section so that the other authors also take care of the fact that there are people all over the world reading TCJ.

b) When printing listings or tables, you should ABSOLUTELY use a font with fixed pitch, like Gothic, Courier or any Line Printer style. Also take care of correct indentation where it is useful. I was very disappointed about the way my bitfunction tables were printed in the last TCJ issue (#63, pages 31 & 32), and about the way my demo pascal program was printed! This way, I agree, pascal programs are NOT easily readable and understandable! Please prefer using a smaller font before deleting space lines in program listings! And if you must print a listing with a fixed pitch font that allows for only (say) 50 characters a line, please inform the authors about that and they will care about this restriction! (In some cases, proportional fonts are also usable when the tables are correctly indented, for example TCJ #64 p.25 the first table. The second table or the dump on p.13 would have better been printed with fixed pitch!)

Then I have something to mention concerning the Small C discussion. If you are looking for the newest Small C compiler for CP/M, be aware that there is a version 2.70 of 30 October 1986, also by Scacchitti. I have it (assumed to be complete) in two ARCs which contain the programs resp. the sources. However, I never tried it out. At least for the CP/M community, there are few problems concerning the availability of a (reasonable) good C compiler since the High-Tech C software was released to the Public Domain (still with copyright, but for free use). However, I didn't try that either. All the larger system also have good C compilers (Unix, GNU) in the Public Domain. The only problem which remains is the portability between those different C versions...

One of the letters in the Small C introductory article also covered software development for the 68HC11 and the 8048. The letter was from J.G. Owen, and he mentioned he had designed and built an emulator for the 8048 by himself. I think this would be a very good hardware project to be printed in TCJ - an article (series probably) about how such an emulator is working, how it's designed and built up. At least I would be glad to have such stuff in TCJ.

Then I have something that you could (please) **PASS ALONG TO RON ANDERSON**. Since the stuff also covers general items, you should read it, too. Or print it in Reader-to-Reader:

Hello Ronald,

at first I want to welcome you to the TCJ writers! However, there are some statements in your letter that I must answer too (you already mentioned you would take needling a bit...).

You are right when saying that the 8250/16450 is generally the same as the 6850 (or the 8251 for the Intel world). However, there is a slightly difference already in the hardware: the other chips WORK. The 8250 and his successors have several design bugs (for example, they don't handle hardware handshake by themselves, and there are status flag errors after writing to the transmitter register), which only proves once more that IBM made the lousiest computer architecture from the worst parts available. Chips like Zilogs Z80-SIO or the Z-SCC never had such bugs (that's why professional I/O boards always use these chips). BTW, your handshake mechanism did work because you stopped the 6850 - vice versa you would have run into much troubles!

So you transferred the data through serial wire to a PC, and wrote it to 1.44 MB floppies. That's another detail: 1.44 MB is also IBM's fault! With optimum formatting, a 3.5" HD floppy holds exactly 1.76 MB which is 22% more! Why? Simply because the folks at IBM are nuts! BTW, this applies to all IBM PC disk formats: the 360k could be 400k, the 720k could be 800k, and the 1.2M could be 1.44M - I did not calculate yet the real capacity of an ED disk, but it will be at least 3.52M (IBM has 2.88M). This all within all specifications for the IBM 3740 disk format. Do you now understand that we are somewhat emotional about the PC architecture?

Then you describe that the conventional way using the appropriate DOS functions didn't work for the serial communication. That's what thousands of others already experienced before, and leads us to the software side of the PC. What purpose, do you think, has an operating system? Is it for the programmers to write programs that directly access the hardware? I think not: an OS should be the interface between ANY hardware and some software, using well-defined interface conventions. Only this way real generic software is possible! Unfortunately, the PC BIOS and MSDOS are so bad pieces of software that already in the early days people directly accessed all hardware because else their programs wouldn't work. This exactly is the cause why today we have the compatibility problem! There were other (better) hardware designs out there which also ran MSDOS (for example: DEC Rainbow, or Victor Sirius) but were not compatible - so the direct-hardware-accessing software would not run on those machines.

So your approach throws us back to the stone-age of software development. I think today we should at least TRY to write generic software! (What will your program do with an I/O card which contains Z-SCC's instead of the 8250? Using at least the BIOS interrupts, it would still run. Using direct hardware access, it will crash!)

The same applies to your idea of directly writing to the screen memory. What if (for example) we are sitting in front of a terminal, which is connected to the PC by serial interface?? What if we have a video board which is not directly memory mapped (that are the better ones)?? Again, at least BIOS interrupts should be used to maintain a minimum of portability!

I should mention that we must generally distinguish between software development for embedded applications and software development for standard platforms. In an embedded system, you can of course freely access all available hardware the way you like. This also applies to people selling turnkey PC's with special applications which won't be used for anything else.

Concerning programming languages, I agree that you can easily translate any Pascal program line-by-line to C (see my comments at pages 41 & 42). However, this does not apply vice versa. So I would like to repeat my statements already printed there (if you didn't read them yet, do it now).

I absolutely agree (without any "but"!) to your comments concerning program sizes. Yes, the neat small utilities which often are somewhat specialized are much more useful than today's blown-up PC software packages.

I guess I also have needled a bit. Hoping it wasn't too much... Greetings, Tilmann Reh

The following is for JW Weaver. PLEASE PASS ALONG and also read it:

Hello JW, (what does "JW" mean?)

thank you for including my suggestions (and Bill's, of course) in your column. I am eager to see the reactions of the readers, and the project forum we created this way.

I would like to see my attributes changed in the Regular Contributors list. The MS-DOS disk emulator is only one of my programs! My general activities can be described as: - Modular computers, especially Z180 and Z280 and based on ECB bus; - GENERIC CP/M-3 (-Plus) software; - embedded control with microcontrollers. Please change this in the list.

For the new project lists, I suggest to divide each list into software and hardware parts. This will result in a total of six lists where the readers easily find the topics they are interested in.

I already have some projects:

Software, finished:
- Generic MSDOS disk emulator for CP/M-3
- Generic MSDOS disk reformatter for CP/M-3
- Generic directory utility for CP/M-3 (4DOS-like)
- Generic harddisk PARKing utility for CP/M-3
- ARC V2.1 which runs with both CP/M 2.2 and 3

Software, needed:
- Generic SCOPY for CP/M-3 (already asked R.Friefeld)
- Generic GOOD (!) Backup software for CP/M-3 (I already have some ideas, but not enough time)
- Generic CHKDSK utility for CP/M-3 (checking the validity of the complete directory)
- Somebody please BUGFIX the disk utility DU90! (there are several hard bugs)
- Turbo-compatible Pascal compiler (maybe very slow!) which generates good optimized code. (I would like to do the final compiling of my programs with such a compiler!)

Besides this, I am strongly searching the author of the PMARC program package (containing PMARC, PMEXT and some utilities, but only poor documentation since that was originally written in Kanji). This is a program comparable to ARC, but with much better compression (about like LHA under MSDOS). The author is named Yoshihiko Mino and seems to be active somewhere in the Japanese MSX scene (an OS standard similar and roughly compatible to CP/M). Until now, all my tries to get any contact to Japan or to trace back the way of the program were unsuccessful. Perhaps anyone of the TCJ readers is able to help me?

OK, that's enough for this time. Let's wait and see what the readers will have to offer for the "new" column.

Greetings, Tilmann Reh

Last item: Remember us talking about a small serial network? Now Rick Rodman has taken up this and wants to do something like we already thought of. This is good luck! The following text also goes to Rick (so you don't need to forward it to him):

Hello Rick!

I was very glad to read your column in TCJ #64. Some time ago, me and Bill Kibler discussed some small computer network (mainly for data exchange) based on the standard serial interface. Since you now are really planning to work in that direction, let me tell you some ideas:

The standard RS-232 has some great disadvantage for building such a network. Since it's defined as one-transmitter one-receiver, you still have to plug and unplug cables if you want to transfer data from or to various computers.

However, the serial principle is basically the most portable one (see Bill Kibler's "Next Ten Years" article). So I would suggest the following: For the line definition, we should use a RS-422/485 interface (the only real difference is the allowed number of transmitters, and tristate RS-422 drivers can also be used for such a project). Now where to get RS-422/485 interfaces for all our computers? There are small interface modules available from Newport Components which convert from RS-232 to RS-422/485 without external components. These modules are available at acceptable prices even in low quantities (at least here in Germany). We might also build such converters by ourselves (and/or publish them in TCJ). It's not much expense. However, care should be taken with protective grounding. Most computers are grounded, so there might be ground loops through the network, or even heavy currents if there are different grounds. When needed, we should have a circuit at hand which is opto-isolated. Again, Newport also sells opto-isolated converter modules.

For the software side, I don't know details about the commonly used network protocols. But I guess that all those protocols have far too much overhead for our purposes. It's just that they are designed for another purpose. I would prefer the definition of our own, tiny and effective network protocol (maybe based on one or more of the common protocols). We don't need much functions - in my eyes, opening (or creating) files and transferring data blocks will do. Of course we need some basic administration functions such as getting the free space on the target disk, transferring file names, date stamps and attributes, etc. But I'm sure we don't need such large protocols like TCP/IP or NetBIOS.

Greetings, Tilmann Reh

So that's all for today - okay, it's already some pages again.
Greetings (& Merry Christmas!), Tilmann

P.S. I just received a message from Rob Friefeld indicating that he is willing in enhancing SCOPY for generic CP/M-3 support!

Well I decided to forward and include all your comments in reader to reader. As to what B.S. and B.A. are, that really got me by surprise. B.S. is for Bachelor of Science and B.A. is for Bachelor of Arts. The B.S. is what you get after four years of college if it is mostly science oriented. The B.A. is the same but mostly liberal arts course work (a history degree versus a chemistry degree (the B.S.)). The G.R.E. is what you get if you take a test to graduate from our High School or equivalent (the 12 years of fundamental schooling), so that you can enter a college program.

I know college systems are different in Europe, but I guess I was mistaken that BS/BA were universal terms in academia. What's your education honors in Germany?

To TCJ's printing problems, I like to blame PageMaker 4. I have had considerable trouble getting the charts to work as stated in the users manual. I follow all the steps, use the proper fonts, and still the program shuffles them around. I have the latest version of PageMaker (the page formatting program I use) but am afraid that they just added new bugs and not fixed the old ones. I hope the new version fixes this problem, as I currently spend two or three hours on charts, and still get a messed up display.

Lastly, Rick and I have had similar conversations about networking as I had with you. I have a copy of a TinyTCP he is playing with and it looks like a possible solution. What we need now is some time to play with it.

Thanks for the comments! Bill.

Dear Bill;

In the Never-Ending-Search for parts sources, especially new IC's, there are two relatively new developments. A few of the electronic distributors have started taking small orders over the phone and accepting payment by credit card. And two IC manufacturers will also take phone orders and credit cards. Note that for common parts on small orders, the prices may seem high. For new or odd parts, they may be your only source.

Electronic distributors are the companies that buy from the parts manufacturers and sell wholesale to the equipment manufacturers. Before, you were required to have a resale permit and be able to open an account with them to purchase parts. I have a resale permit and pay cash when I buy parts, but several distributors wouldn't accept cash or sell to me unless I filled out a four page credit application to open an account. I was told that was the only way they could put me in their computer.

Note that distributors are not like stores. They stock only what their (largest?) customers are currently ordering. Many of them specialize in selling just IC's or just resistors and caps. It always helps to know correct part numbers when you call.

DISTRIBUTORS

ARROW CATALOG DIVISION, 1-800-932-7729. Manufacturers: AMD, Intel Motorola, TI, etc. Full product line appears in catalog, but probably doesn't stock everything. Catalog they sent me recently was dated 1988.

BELL INDUSTRIES, 1-800-289-2355. Manufacturers: Analog Devices, Maxim, National Semi, etc. They carry a complete line also from batteries to wire.

WYLE, 1-800-414-4144. Manufacturers: AMD, Intel, Harris, Motorola, TI, etc. WYLE specializes in semiconductors and computer systems.

MANUFACTURERS

MAXIM, 1-800-998-8800, Ext. 6380. Products: Analog switches, power supply supervisors, video amps and switches.

DALLAS SEMICONDUCTOR, 1-800-336-6933. Products: Battery-backup controls, clocks, rams, and VERY fast 8032 microcontroller.

OTHER STUFF

Don Lancaster, author of the TTL / CMOS / Active Filter Cookbook, has a column called RESOURCE BIN in Nuts and Volts where he talks about resources for electronics, from tools to parts. A book of his columns is available from his company, SYNERGETICS. Call 1-602-

428-4073, Don may answer the phone.

By Dave Baldwin

Thanks Dave, good information and thanks for checking out the references and the next information.

XT MOTHERBOARDS, CHEAP CONTROLLERS

I've been keeping an XT clone running for testing programs and some hardware. I changed a couple things and went to edit my menu file. The keyboard kept screwing up so I finally took the thing apart to check it out. I dug out an old test rom and plugged it in and it kept showing me random memory errors and failures in the peripheral chips.

I changed the memory and a couple of other chips, but random errors kept occurring. I've been collecting clone XT motherboards at swap meets for about $2 a piece so I got one of them out and put it in the machine. It passed all the tests perfectly so I put the chips from the first motherboard in and ran the tests again. It still passed all of the tests. I tried everything in the first motherboard again and I got intermittent errors again.

Well, I remembered that some of the early PC clones had some problems. I had all these XT boards I had collected so I tested the rest of them to see which ones to keep and which ones to put in the parts pile. I noticed that the only ones that passed all the tests were DTK boards. None of the other brands worked. Then one of the DTK boards didn't work. I took a close look at it and compared it with a working board and noticed that half of the bypass caps were never installed. After I put in most of the bypass caps, it also passed all of the tests.

You can recognize the DTK boards because between the first and second edge connectors is the DTK logo and the words "COPYRIGHT : 39784" with "PIM-TURBO" below that. These are 8 Mhz 'turbo' boards with a software or hardware turbo switch. The latest rom version I've found is ERSO/DTK 2.39.

I had been collecting these boards because they're cheap, have all common parts, and information on them is available. The only special part on the boards is a 256x4 prom (74S287) used for decoding. Most of the boards were discarded for upgrades or because someone couldn't get them to work. One of them was missing a jumper that enables the keyboard. Others came with a full 640k of memory.

The advantages of old XT's for projects are: **They are cheap and use common parts. Up to 640k of ram and 48k of rom on the main board. Power supplies, plug-in cards and cases are available. You can use your desktop PC to develop the software.**

Since lot of projects don't require a better or faster computer, I figure that this is one of the cheapest ways to build a system.

Of course, faster is better!? Using NEC's V20 speeds up an XT by 10 to 30 % over an 8088. XT's ran slower from rom than ram because the fastest eproms available then were only 200 ns. They required a wait state to run at 8 Mhz and ram didn't. You should be able to speed up rom access by using 120 or 150 ns 2764's and cutting the trace for the rom wait state.

Of course, there are other boards that actually work. I have some genuine IBM boards that I haven't had time to test. A friend of mine had a Faraday motherboard that included serial ports and a printer port on the motherboard and ran at 7.3728 Mhz which was divided by 4 to generate the baud rate clock for the serial ports.

Dave Baldwin, 12/26/93

Well Dave that fits in nicely with our new column on PC/XT's by Frank Sergeant. It also supports my position that in many ways your are better off just buying something other than a PC for learning and hacking. If you didn't know much about hardware you could spend plenty of money and time trying to use these machines and never really know if it was something you did or something wrong with the XT board. My advice for beginning hackers is to make sure it works completely before you do any modifications to the system. Starting out from a known base is always the best way to start.

Thanks Dave for the comments and your article on reset circuits. This should help any would be hackers. Bill Kibler.

Dear Bill,

Once again procrastination and excess task mismanagement has taken its toll on my keeping-up with the subscriptions. Here's the check for another two years, keep up the good work.

I'm currently moving ahead on the YASMIO board (albeit slowly) and hope to have a working model available in a month or two. I now know some of the frustration Paul Chidley must have gone through with the YASBEC! I'd write an article on the project, but I cannot claim full responsibility for the design or the shakedown and it uses the hated GAL for addressing. It may not need it now in its pared-down state, but it's there now and there are a few things that it may be handy for and it does save a LOT of real-estate on the board!

I look forward to the next two years and hope to see some more articles like Terry Hazen's SCSI Programmer and of course Jay's column which I live for each month.

Regards, Jim Thale.

Thanks for the status report Jim. We sure would like to see that article with GAL information and all. It doesn't matter to our readers who did what, as long as you tell us how to get one of the boards (and GAL) so we can build one ourself.

As to the GAL in an article, my position and that which most reader agree with, is tell us how to make the GAL, but also tell us how to build it using TTL if we so decide. My position is that giving GAL/PAL programming code doesn't explain anything about what the logic is really doing. Since we are trying to teach our readers about logic, trouble shooting, and yes programming PAL/GALs, some *explanation of what it does and why is needed. The simplest way to achieve that is by simply showing the TTL circuit that is replaced with the one GAL.*

Most important Jim is not getting bogged down in the choice of one device when the whole is what we are after. In this case the whole article and your board is what is important. Thanks again. Bill.

Dear Bill,

Time has passed faster than I had expected. A couple of weeks ago at the Forth Day, you said that you had Market Place adds at $100 for 6 issues. I have prepared a simple ad which measures 2 x 3 and should fit nicely. I am enclosing the add copy and a check for $100.

I like what you are doing with TCJ and I gather that it is beginning to grow. I would be happy to consider putting the description and circuits for the WISC/16 in a series of articles. It would provide all of the information required to wire wrap the system. Only simple TTL 74xx chips are required.

The first article might be an introduction to building in hardware the system that Forth emulates, then the sufficient circuit to interface with a PC host, followed by the stack design, the ALU design, memory access, and finally the microcode RAM in successive articles. After the PC host interface is wired, all further steps can be checked with Forth test routines as they are wired. I use the same tests to check our assembled and tested version of the same design. Who said you can't build a processor at home. It is fast too!

I will look back over the several issues of TCJ I already have for article ideas and would be interested in getting those back issues we talked about.

I look forward to hearing from you in due time.

Sincerely yours, Glen B. Haydon, M.D.

Thanks for the Ad Glen, and I sure enjoyed our talk at Forth day. I have included your ''levels of Forth'' elsewhere in the issue, with a few thoughts we had about starting beginners on the right foot.

As to the WISC/16 I am very much still interested. The idea of our readers being able to build their own super fast Forth co-processor for a PC is exciting. If I remember right it was faster than the NOVIX (10MIPS at 5MHZ). That is just amazing for a hand wrapped board. The real beauty of the project is only simple TTL devices are used!

Seems I just found your message about the back issues, oops, sorry for the delay, but hopefully with my new job and all, I will be catching up on the business side of the magazine.

Thanks for your note and all the years of supporting Forth. BDK.

I just got issue #64 and would like to comment about ''SMALL-C?''

I learned to program in Fortran in a college class and then taught myself Basic (spaghetti code) and assembler for Z80 and 6800. Yes I know one can write in Basic with some structure, but it usually doesn't happen. Then as a CS student I learned Pascal, assembler for 8088, and C. I presently write mostly in C, Z80, or 68K assembler.

I strongly agree with Tilmann Reh. Either use pseudo code or a Pascal type language so that the source can be most easily read and understood.

Thanks, William Winton.

Thanks for the comments William. I hope others like you send in notes about their choices. I might suggest you read the "levels of Forth" article to see if your choices agree with Glen's idea of learning languages for your level of needs and skills. You didn't mention the end results of your programming, but since you do assembly, it must be near the hardware level, is it? Bill K.

Articles Neded

We need articles on subjects that are of interests to our readers. Those interests now span small and older eight bit systems, through the obsolete IBM PC/XT style of computers.

The subject matter of interest are mostly those which explain and teach readers how to perform intermediate and advanced improvements and modifications to their systems.

All of TCJ's readers are not intermediate in skill, many are beginners. Articles need to take any reader of any skill level through your project, as if they were begining on this subject for the first time.

Areas of current interest are using older and obsolete systems for new embedded control situations.

Embedding operations in ROM and running the entire operations for remote sensing over a telephone line would be a great article of interest to our readers.

First hand reports on the history of early and classic systems is always a topic which our readers enjoy.

Projects which use surplus parts available from current vendors, showing how to debug and develop the needed knowledge of the used system, is something of interest to our readers and advertizers as well.

Short reports on projects that are currently under way, belong in our Support Groups section, where letting others know of what is being done has become a major focus.

Send your letters to:

The Computer Journal
P.O. Box 535
Lincoln, CA 95648-0535

Regular Feature

All Users

AMSTRAD

The European Beat

by Helmut Jungkunz

Sugar for Computers

Yes, you've known it all along, computers can be really sweet. But - did you ever connect sugar with computers? No? Believe me, this connection was one of the best ever for Z80-based machines.

Maybe I have to be more precise: Alan Sugar is in full truth the renowned Alan M. Sugar Trading Company, who formed a big name: AMSTRAD.

Actually, the original AMSTRAD computer was supposed to be a clone of the C64 with a 6502, which, by luck, didn't work out in time, so they took a Z80. The funny thing about that machine was the fact that it was designed "outside-in", meaning, the housing was ready, but the circuit board wasn't! But back to our story.

In 1984, AMSTRAD shook the market for CP/M Computers for the first time: The Colour Personal Computer (CPC) 464 was cast into the home computer market, then offering not only a perfect game console, but also a very effective and powerful BASIC, the Locomotive BASIC, in the ROM of the computer. This BASIC had been written by the same people, who had done the famous BBC-BASIC for Acorn. A built-in high-speed cassette tape loader managed to outrace the floppy-drive of the Commodore C64 (naked,no speed-DOS)! Thus, for very little money, you had a 4 MHz Z80 machine with an extension bus to take all sorts of peripheral devices, the most interesting of which without question was the external floppy controller. This floppy-controller came together with CP/M 2.2 and the typical AMSTRAD 3" (Yes, 3", not 3.5") disk drive. The reason for this format was, that the decision for the new disk size hadn't really been taken and the odds were fifty-fifty between 3" and 3.5". Since the TATUNG "Einstein" started to use 3"-disks, AMSTRAD followed this sign together with several industrial developers. Nobody had an idea, what dramatic problems this decision would cause.

Anyway, CP/M 2.2 at hand and the ability to use various disk formats made the CPC look really good. The Standard IBM-8 SPT SS40 format with 153K capacity was typically available on most CP/M machines. On the CPC this was nice, but useless, if you only had one disk drive, normally a 3" drive. But the 3" disks themselves were not all that bad. In fact, they proved to be the most robust for shipping by mail and handling was next to fool-proof, since a slide able write-protection was integrated into the disk design and bending a disk by accident was completely impossible.

CP/M loading on the CPC is done in no time at all. A slight disadvantage was only, that of the addressable 64K, CP/M would use 16K for itself, thus leaving a meager 39K Transient Program Area for load able code. Now, you might ask, what good is a CP/M machine, that can barely run WordStar and not at all dBASE or Multiplan? Patience, we'll talk about that later. For the average user, public domain software would have been ideal. So why had all these other computers have to be in such awkward formats like 8" or 5.25" ? Fortunately, soon some people noticed, that you could hook up not only a second drive, but that the disk drives could be anything, they would just be of the same logical format (how the operating system "sees" the disk), despite the different physical format. That was the time, when adding a 5.25" drive to the CPCs became a regular thing.

After a short while, SIG/M software had found its ways into AMSTRAD CPCs. The need arose, to meet the demands of the software market for more TPA. VORTEX, a company from the Heilbronn area in Germany, soon came up with a RAM-expansion card to plug into the CPC. And into the CPC it was. You had to open the computer (oops, warranty) and pull the Z80 and the complicated AMSTRAD Gate Array out of their 40-pin sockets and plug them into matching sockets on the RAM-expansion card. From there, flatwire cable arrays went to 40-pin plugs to take up the vacant spots on the computer circuit board. All in all, a horrible construction, leaving many customers next to helpless.

Another problem showed shortly after. The 50 pin expansion bus of the CPC is not buffered and has a very small capacity of driving any load. So, the additional circuitry of the VORTEX expansion RAM really brought exciting new effects to the world of Amstrad users. One of the most likely to happen was the so-called "CAT-syndrome". This has nothing to do with anybody's pet animal, but with the BASIC command "cat" used for displaying the contents of a tape or disk. When the computer starts, a "ROM-walk" is performed, initializing all ROMs. The external disk controller ROM (inside the controller housing), when installed by this ROM-walk, simply returns and the BASIC ROM issues it's READY. In the CAT-syndrome case, this READY would never or seldom appear. The message would flash in and out, but never come to the point where it could put out READY. Solution: the ROM in the controller was replaced by an EPROM, which is a bit more tolerant.

These problems with the VORTEX expansion-RAM-card soon lead to the forming of the VORTEX User Group, which, after a short while, turned out to be a clever gimmick pulled by a salesman to hook CPC customers. One of them, a certain Peter Immerz, had the intention of founding his own CPC Club. He called it, in accordance to its German brand name "Schneider", the "Schneider CPC User Group". A couple of people immediately joined this group, among them Andreas Kisslinger and me. After a rough period of establishing, our club started to work fine. Andreas had written an excellent program to install a different DPB (disk parameter block) in the CP/M, so that 80 Track drives, hooked up as B: could be used with all the 80 Tracks possible. Since the BIOS didn't support double sided disks, flippy disks were quite common. You know, the kind of disks with two index holes and two cutouts that you could flip over and use the other side.

Shortly after, the CPC 664 was introduced by AMSTRAD. That model contained the dreaded 3"-disk drive already and had no cassette drive. This pushed the market for disk oriented 3"-CPC games quite a bit. The standard CP/M software was available from Markt & Technik in Germany, some of it only with the addition of the VORTEX-RAM-expansion, like dBASE or Multiplan. WordStar (3.0) could run in the meager native CP/M TPA. Programmers started to hack all sorts of Public Domain tools and some even got fully translated into German with printed manual and all files on 3" disks! The users were so busy, they were absolutely shocked, when shortly after, the PCW 8256 was announced, the "Joyce", as it was called during the development period (after the secretary). By the way, the CPC's nickname was "Arnold".

This PCW had frightening new things to offer: CP/M Plus operating system (3.1 - the "repaired version"), a compact design with monitor, computer and disk drive, all in one. A comfortable keyboard made clear what the machine was aimed for: professional and semiprofessional office use! In fact, the word processor included, was one of the most powerful available for CP/M machines then. The reason for this was simple: the Locomotive people (the ones who had written the BASIC, remember?) had developed the Z80-based word processor "DIAMOND" some time ago that used to run for .... US $. There didn't have to be many changes made to adapt it to new hardware. To complete the office design, a small and handy matrix printer was included with the PCW. A clever marketing thing - to sell computers like Hi-Fi stuff - a typical AMSTRAD idea. This concept was adopted by the German Schneider GmbH, who then started to sell AMSTRAD computer products and offered repairs from their existing net of electronic service centers.

The next surprise was the fast advent of the CPC 6128, a twin of the 664 concerning CP/M 2.2, but a little different in its BASIC, able to run in different banks through BANKMANAGER, an AMSTRAD program included with that particular computer. On top of it all, a perfect implementation of the AMSTRAD CP/M Plus came with it, possible through the 128K RAM memory with bank switching, but still allowing the typical CPC firmware calls through the XBIOS calls.

Digging into the machine.

One of the great disadvantages of the AMSTRAD CP/M 2.2 terminal was the incompatibility with practically all existing terminals. Only CTRL-L (Formfeed) as ClearScreen command turned out to be useful. An interesting feature to modify screen colors by sequences containing nulls and a true windowing function with a starting byte of CTRL-Z drove everyone crazy, when trying to run a Televideo-type program, like from an Osborne or Kaypro terminal. Those computers use CTRL-Z for ClearScreen and their programs are full of single byte CTRL-Zs, followed by HEX 00. This produced a beautiful window size 0 on the CPC. Great.

Now, finally also on the CPC - a common terminal. The AMSTRAD CP/M Plus supports a subset of the VT52 terminal, pretty much the same as the Heath/Zenith terminal. The main difference in the two is the way the screen is erased. The Heath/Zenith and AMSTRAD terminal use ESC-E to erase the screen and ESC-H to home the cursor, whereas the real VT52 uses ESC-H to home the cursor first(!) and ESC-J to erase to the end of screen (EOS).

So, where did this leave us users now? We now not only had three different BASIC-computers or three different CP/M machines:

BASIC		CP/M 2.2	CP/M 3.1	DS Drives
CPC 464	Tape/disk	w. extra DDI-1	no	no
CPC 664	disk	yes	no	no
CPC6128	disk New BASIC	yes	yes	no
PCW8256	disk	no	yes	yes

As mentioned before, in Germany, the former Hi-Fi company SCHNEIDER had opened an agency for AMSTRAD and used their own brand name, so the AMSTRADs are also often referred to as the "SCHNEIDERs" in Germany and Austria. The slightly different keyboard layout and more uniform colour had prepared ground for SCHNEIDER. Further, the English CPCs and PCWs were different from keyboard and connector layout, so the peripherals had to be different for both versions. When dk'tronics in England came up with RAM-expansion cards for the CPCs, they wouldn't fit on the German Schneiders. On the other hand, the VORTEX-RAM expansion card would fit in the British CPCs, but the British didn't know about it. It is simply a fact, that computer development flowed in one direction only, that is, from USA and England into Germany and seldom the other way around.

That is perhaps the main reason, why two totally different worlds developed. The dk'tronics RAM-boxes initially suffered from weak hardware design, but came with a nice tool to run either a big CP/M 2.2 with large TPA or a RAM-disk, or both. Also, CP/M Plus was possible on the CPC x64s. This was achieved by sticking closely to the AMSTRAD standards. The VORTEX RAM-expansion card also allowed for a large TPA in CP/M 2.2 (62K!), but the BASIC supported was quite different from the BASIC 1.1 of the 6128, which was supported by the dk'tronics hardware. Like the dk'tronics, the VORTEX RAM-card allowed for a RAM-disk of full-blown 444KB net capacity (the actual 512KB mapped the first 64KB to the parallel bank address and was used for bank switching).

The only real advantage of the VORTEX card was the separately loaded BIOS-file $OSC.SYS, that couldn't be loaded from the system tracks because of its size. This soon inspired the cracks to disassemble it and try to comment what they found. And brother!, what they found, they didn't like. The code was filled with redundant loops and fuzzy jumps. Much of the stuff could have been eliminated by using labels and constants. So, whatever could be clearly identified, was partly rewritten more efficiently or rather patched for better action. Also, some new features were added, like disk parameter block manipulation for new drive type support. Double sided disks with either double-stepping or none-double stepping could be used as well as single-sided types.

This whole "new" BIOS was released to licensed owners through our club and equipped with the programs SETFMT to set, change and/or format various disk formats. Also, the program to initialize the RAM-disk was rewritten from scratch and much better in my opinion. The new tool was called RAMINIT and made use of the extended user areas of CP/M 2.2. With special tricks, one can access more than the usual 16 user areas, actually up to user 32. Here, the author (Andreas Kisslinger) deposited a special string that was checked for existence, and, if found and ok, then the first 16 entries of the directory were checked. Where the original tool RAMDISK formatted the RAM-disk no matter what and never checked anything, RAMINIT presented you with the choice to do so, if the afore mentioned checks were successful, otherwise it would automatically format the RAM-disk. Since German users needed "Umlauts" ,i.e. a,o,u, diarhesis, a way had to be found to install a translation. Since there were several solutions possible, there were different solutions.

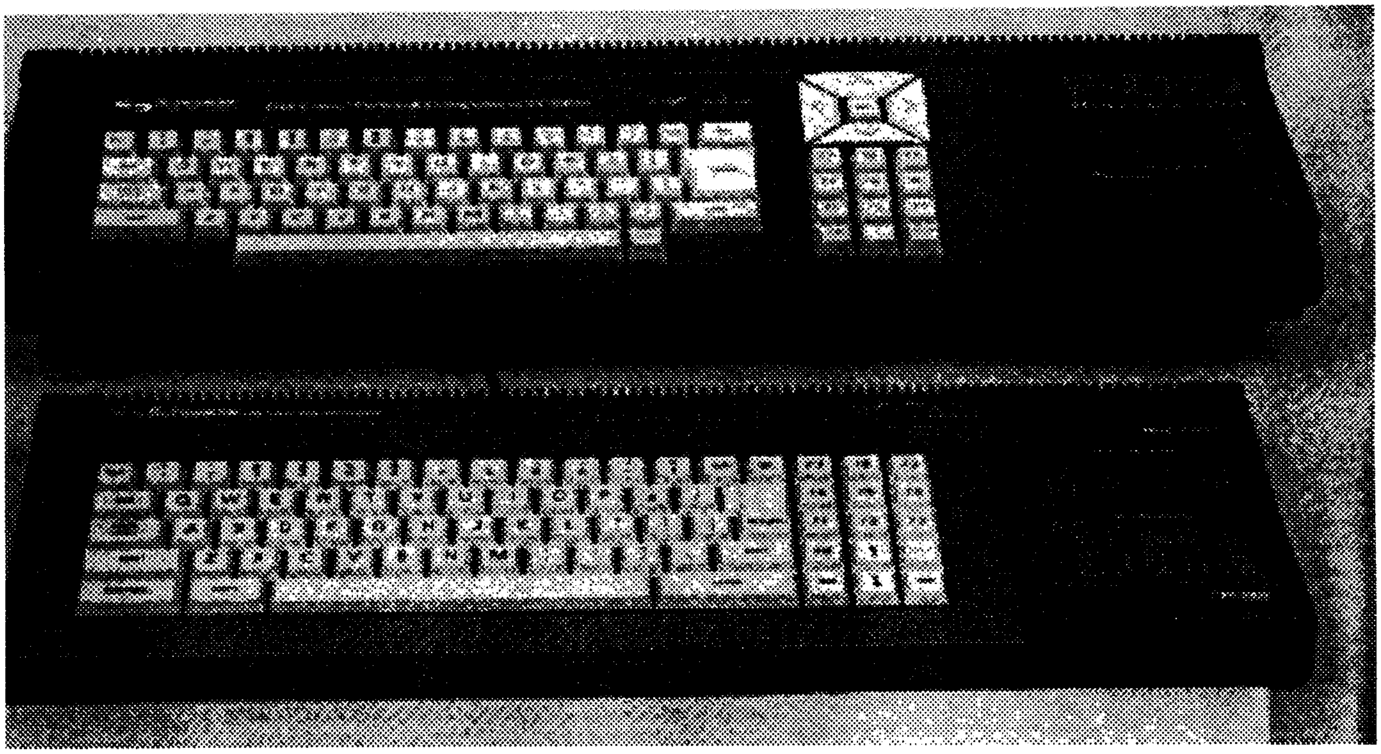
Markt & Technik, the distributor of WordStar, dBASE 2.41 and Multiplan for the AMSTRAD line, chose to implement a patch into the program, that demanded a VORTEX RAM-card and wouldn't run the program without it. In the beginning, everybody thought that would be okay, until other RAM-cards appeared on the market. Then, the patch would have to be removed manually, usually my job, since most users were unable to do it themselves. The other way to install "Umlauts", was a program written by one of our club members: DEUTSCH.COM. You would then call the program with an argument of either "ein" or "aus" to toggle on/off. The same guy (Helmut Tischer) also wrote a whole bunch of tools for the VORTEX CP/M and regular CPC CP/M 2.2. There was a keyboard patch program, several disk-tools to read the sector I/Ds of any disk, patches to alter the step rate of the disk-drives and many others. He adapted Z80DOS and wrote a CP/M-replacement to support a realtime clock and multiple disk types (BANKCPM).

Of course, there were also others who wrote programs and tools for that special unique environment, even commercial ones. The most famous team consisted of Peter Hoepfner and Detlev Gunkel, who among comfortable user interfaces for disk handling and copying finally even came up with a special CP/M Plus for the VORTEX RAM-card. But that was a long way to go.

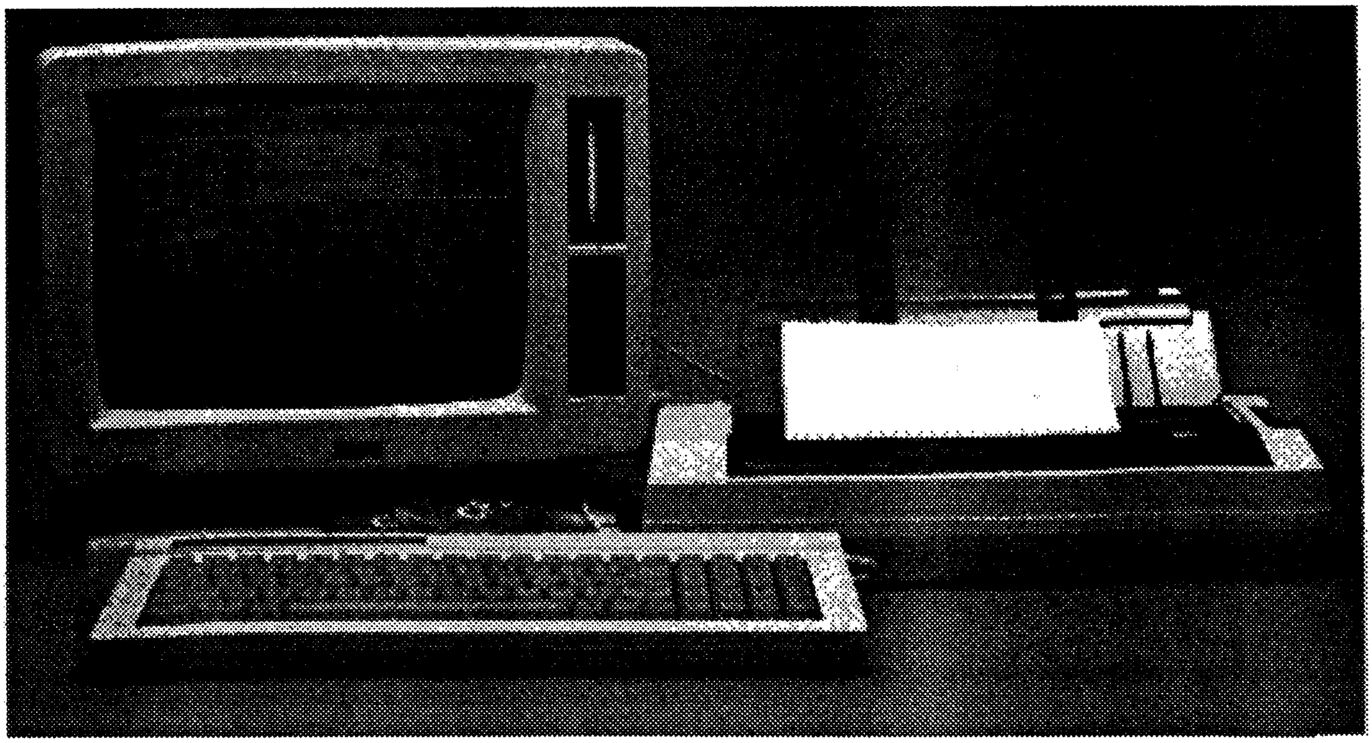
In my next article, I want to cover other developments for the AMSTRADs and modifications made by our club members. I hope you enjoyed reading this. Let me know, if you have any specific questions on AMSTRAD computers. I will try to combine information with historical anecdotes in order to give you a round picture.

Helmut Jungkunz, "Virtual" ZNODE #51, or CompuServe 100024,1545

A full CPC System



CPC 664 and 6128



Schneider PCW8256 'Joyce'

Regular Feature

Intermediate Users

PC/XT Architecture

PC/XT Corner

by Frank Sergeant

Day-Old Computing

I fell in love with the original IBM PC the first time I touched it. When I was a child, my daddy explained how to make someone enjoy a beating with a rubber hose: beat him with a hammer first. The PC was the rubber hose. Its keyboard felt great. Its green long-persistence-phosphor TTL monitor was beautiful. Its floppy disks were (almost always) reliable.

The hammer I'd grown up with was the Radio Shack Color Computer, with its lousy keyboard, flaky disk, and hideous display. It had a heart of gold inside, the Motorola 6809, but was physically uncomfortable. The PC has a heart of lead (or worse), but I guess clothes make the computer.

So, now, fickle readers and publisher, you'll accept the PC/XT just because it is old and abandoned? Well, I have no honor either, since I'll fall for an Intel 8088 if it is dressed in a pretty keyboard. Let me express some disclaimers. I do not like the 8088 processor, with its special purpose registers and cumbersome segmentation. What I like is that the PC is cheap and universal (and let's not forget the keyboard). I like it because of the Eli Whitney syndrome: it has mass-produced, 2nd sourced, interchangeable, cheap parts. It has an "open architecture." If you oppose the PC because you prefer the 6809 or 68000 processor families, I agree with you. I have often cursed the 8088 and threatened to go to the 68000 and to hell with compatibility. But, if you oppose the 8088 because you prefer the 8080A (or its Z80 or 8085 spinoffs), well, I think you've missed the point that it was cut from the same mold.

Let's hate the PC. Let's hate it together. But, let's hate it for the right reasons. The PC world is troublesome because it changes rapidly. So many hardware and software add-ons are available that no one can keep up with them all. Before you learn one version of a bloated "application," it's time to install and pay for the upgrade, which requires more memory and a larger hard disk. Is there any way to take advantage of the good features and avoid the bad? Is the PC/XT now a simple, classic computer; easy for beginners to learn with?

Components

The PC is modular. You can buy it piece by piece and put it together. It requires 8 essential components for desktop use:

1. Case
2. Power Supply
3. Motherboard
4. Keyboard
5. Video controller card
6. Monitor
7. Disk controller card(s)
8. Disk Drive(s)

Prices

The original PC came only with floppy disk(s). The PC/XT added a 10 megabyte hard disk. The PC had 5 connectors (slots) on the motherboard for add-in cards. The XT had 8 slots. Other than that, we can treat all of the PC, PC/XT, and clones as more or less interchangeable. The early versions had low-wattage power supplies that weren't suitable for plugging in too many additional cards and power-hungry hard disks. The early versions only allowed 256K bytes of RAM (memory). Soon this was raised to 640K bytes. What prices are we talking about today? I'm going to give some rough estimates. Case $35, power supply $35, motherboard (standard XT clone 8088 running at 8 or 10 MHz) $39 -- maybe less, keyboard $39, video controller $20, monitor (TTL) $79, disk controller $20, disk drive (360K floppy) $20, serial interface card $20, parallel interface card $20, hard disks about $1 per megabyte. Various suppliers often have much lower prices on some of the above.

For example, you might get a TTL video card or 360K floppy controller card for $9. I think you can buy XT motherboards all day for under $39 (with memory?). But, I've now seen 80386SX motherboards as low as about $80, so it might make more sense to get the '386SX motherboard. You can put 512K bytes of RAM on the '386SX motherboard for about $20 (or multi-megabytes for maybe $40 to $50 per megabyte). You should be able to buy a surplus PC for under $150, assembled, perhaps less, used. The problem with buying a used computer is the seller remembers what he paid for it, rather than what it is worth now, but you might still find a bargain.

Printers and Disks

In the PC world, most printers expect to connect to a parallel port. This port is usually included on the TTL (monochrome) video card. If not, it can be added for $20 or less.

The first PCs tried to boot from the floppy disk in the A drive. If this failed, they jumped to BASIC, which was present in ROM (Read Only Memory). Clones do not have IBM BASIC in ROM, so they just report an error.

Processors

The processor in the PC is the 8088. This is 8-bit data bus version of the 8086. Internally, especially from the programmer's viewpoint, the two chips are essentially identical. When the AT came out, PCs moved to the 80286 chip. It will run all the 8088 software as if it were an 8088, except the 80286 is faster and has a 16-bit data bus. Then came the 80386 chip. It, too, runs software exactly as if it were an 8088 or 80286, except it has either a 16-bit data bus ('386SX) or a 32-bit data bus ('386DX), and is even faster. Ditto the 80486, but it and its bigger brother "The Pentium" don't really fit into my "day-old computer" category at the moment.

The PC has an address space of 1M bytes (1 megabyte). The lower 640K bytes are all that are available to DOS and your programs. The rest of the space (384K) includes video RAM (on the video controller card(s) and the BIOS ROM. DOS itself may take up around 70K. Unfortunately, the PC accesses memory in segments of at most 64K bytes. Then again, a lot used to be doable in 64K bytes. The segmentation is a built-in limitation of the 8088 processor and even of the 80286 processor. If you run DOS, it is even a limitation of the 80386 processor.

However, for the first time, the 80386, with special software, allows accessing the memory as a single, huge, flat address space (like the Motorola 68000 family has allowed from the beginning -- I'm not bitter, really).

Video

Originally video cards and monitors came in two flavors: TTL monochrome (TTL for the digital logic voltage levels used in the interface) and composite color CGA (for Color Graphics Display). The TTL displays were and are beautiful. You had a choice of green or amber. The original monochrome card had no graphics capability, but that was soon fixed with the Hercules monochrome graphics card (MGA for Monochrome Graphics Adaptor), clones of which are available for $20 or less. The CGA display was horrible, absolutely unusable, as far as I am concerned. Finally, today, we have almost affordable monitors (under $300 compared to perhaps $79 to $99 for TTL monitors) and VGA (Video Graphics Array) controller cards ($40) which give tolerable color displays. (And, the sky's the limit for big, high quality color monitors.) Several months ago at a CompuAdd sale I bought an amber TTL monitor for $25 and a "paper white" (i.e. monochrome) VGA monitor for $35, and VGA card for about $20. One can't have too much computer equipment lying around, taking up space, absorbing spare cash.

There is no end to what you could add to a PC: sound cards and speakers, CD ROMs, FAX/modems, but let's exclude these from our day-old computer project for now.

Hardware Compatibility

There are three "interfaces" to the PC. The lowest is the hardware level. Compatibility between systems at this level means the same memory address and I/O port locations, the same type of serial chip, disk controller chip, etc. For example, it has been common to handle video by writing directly to the video memory at a base address of $B000:0000 for monochrome cards and $B800:0000 for color cards. Such software only works on PCs that are compatible with the "standard" IBM and clone systems at the hardware level. Most "PCs" are compatible at the hardware level. I would guess the MS-DOS palm-tops are a major exception, along with some name-brand clones. In general, no-name Asian clones are very compatible at the hardware level, and I prefer them to the name-brand clones.

BIOS Compatibility

The middle level is that of the BIOS (Basic Input/Output System) ROM. The PC has a BIOS ROM on the motherboard. As a reader remarked, the BIOS is what knows the difference in the hardware. The services offered by the BIOS have remained very compatible across all versions of PCs for the 8088, 80286, 80386, 80486, from both IBM and clones. At this level of compatibility the hardware doesn't have to be identical as long as the BIOS does it job and presents a consistent face to the outside world (i.e. the software). Thus, if your software makes an "INT $10" call to the BIOS to request a character to be displayed on the screen, your software doesn't care where the video card's memory is located, just as long as the BIOS itself knows where to find it. The problem with the BIOS is it is often much slower than directly accessing the hardware would be.

Various books describe the BIOS services and how to access them. Essentially, you use an assembly language routine to put certain values in the 8088's registers and then force a software interrupt (an INT $10 assembly language instruction, for example, invokes interrupt number 16 decimal. The BIOS inspects the registers to see what service has been requested, does the work, puts some status information in the registers, and returns to the calling routine. Or, your language may provide a higher-level interface to the BIOS for you. Note that a BIOS ROM comes with your motherboard at no extra cost. Every PC clone in the world has a compatible BIOS (well, with minor exceptions, perhaps).

Operating System Compatibility

Running above the BIOS is the operating system itself, PC-DOS, MS-DOS, DR-DOS, etc. Again, services are requested with software interrupts, especially the famous INT $21 which is the main interface to DOS. The DOS then performs its work by making calls to the BIOS. Using it requires you or your software users to have DOS. Also, calls at the DOS level tend to be higher level. The BIOS can read and write sectors, but DOS can build and access directories and files.

Motherboard Tradeoffs

I think it is OK to pay $130 for an old PC/XT (no hard disk). But, if you are putting one together from components, I suggest you pay the $80 or so for a '386SX motherboard in place of the 8088 motherboard (at $20 to $40?). The upgrade ability will be there when you want it, but you can ignore the advanced features and treat it just like an 8088 until you are ready. Some of the earlier BIOS ROMs in PCs didn't allow the addition of hard disks. Most 8088 motherboards are more difficult to add the larger capacity floppy disks and don't allow more than 1MB of RAM, whereas such additions are easy with a '386 motherboard. Under almost no circumstances would I consider getting a '286 system. It's probably unfair, but I had more trouble with one '286 system than I've had with 8088 or '386 systems.

Embedded Systems

What is the ideal configuration? It depends on what you are trying to do with the computer. For embedded applications, you could get by with a power supply and motherboard, replacing the motherboard's BIOS (Basic Input Output System) ROM chip with your custom application burned into an EPROM (Erasable, Programmable ROM). Or, you could add the disk and controller and load the application from disk. A serial controller card could allow access to the PC from any host computer with a serial port. Bill suggests we work up a way to control the PC using the built-in keyboard connector instead of a serial port. You'll probably need to put a custom ROM in the motherboard if the disk is eliminated. The PC keyboard controller on 8088 motherboards is primarily a one-way device. I'll have to look into it further to see if it can serve as our primary interface. I think it will be fun to look into using cheap XT motherboards for cheap embedded systems.

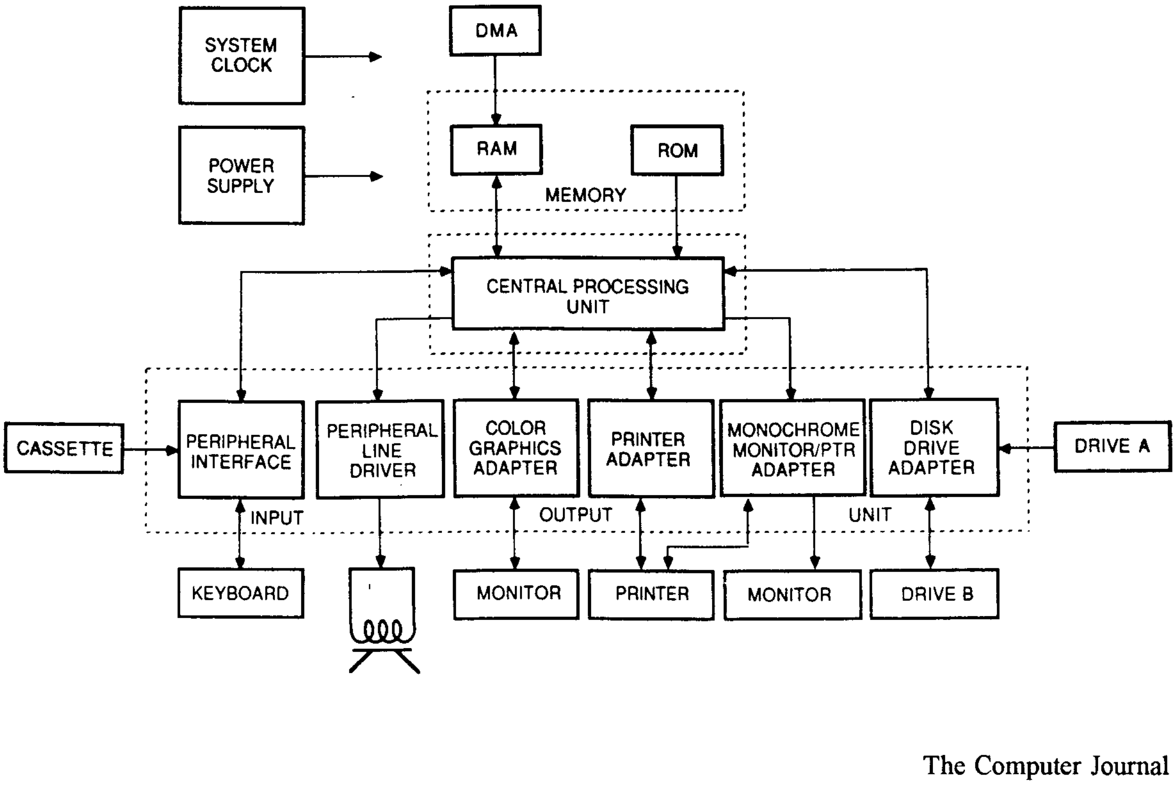
I understand there is a clone DOS available as shareware. I have ordered a copy and hope to report on it next time. Even if this DOS doesn't work out, older versions of MS-DOS are often available for under $20. Of course, we might put Forth into ROM on the motherboard and avoid DOS altogether. If you are already using a PC, it might make sense to use the XT for embedded systems, because you are already familiar with it and because it is cheap. Otherwise, you might be better off using a smaller, lower power board (perhaps built around the 68HC11, one of my favorites at the moment). The lower power board could use cheaper, smaller, easier to backup power supplies and smaller packages.

Desktop Systems

Then again, you might be thinking of a PC as a cheap desktop computer, similar to a CP/M system. There are a bunch of shareware programs that will run on a 256K to 640K PC with monochrome monitor and just floppies (or with a hard disk). For example, there are several WordStar-clones and PCWrite for word processing, and QEDIT makes a nice text editor. There are data base programs (PCFILE) and spreadsheet programs (ASEASY) and drafting programs (DRAFTCAD). There are even CP/M emulators. All things considered, I don't think a PC is a bad choice, even if you just want a simple system to learn with.

Conclusion

There you have an introduction to old IBM PCs. If this is to be a regular column, do you have any suggestions as to what topics to address? Please email or write. I'll save most of my rantings for a separate column.



Small System Support

By Ronald W. Anderson

Regular Feature

68xx/68xxx Support

6800/09 Serial Ports

Here I am again. Bill Kibler says he liked my first column and has invited me to be a regular contributor. I don't know if I can always be at deadline time, but I'll try hard. Perhaps I can get ahead a few columns (build up a comfortable buffer) in case I don't have time later sometime.

68000 Systems

Where were we? I was just re-reading the first column and it occurred to me that I ought to talk about the 68000 systems from Peripheral Technology just a bit. These machines, presently the PT68K-4 and PT68K-5 are being shipped with a ROM monitor called MONK. These are set up to run an operating system called REXDOS available from Palm Beach Software. REXDOS is more FLEX like than SK*DOS. It has been made public domain and is shipped free with the computers. Full source code is available on a couple of disks for a nominal fee (It was $15, I think). The -4 is a 68000 based computer, available in 16 MHz version. The -5 contains the processor on a plug-in board. I believe it is being sold with a 68020 processor. Later it will be capable of upgrading to a 68040 or whatever without throwing away the whole motherboard. I have a -2 next to me at this moment and a -4 at work. The -4 is capable of working with high density floppy drives. The -2 is not, since it contains a Western Digital 1772 floppy controller.

If you want to run SK*DOS, when you order it from STAR-K Systems you will have to order the HUMBUG ROMs for the PT. SK*DOS has a multiple directory system and it codes a time as well as a date when it writes a file to disk. In addition it has I/O redirection via the redirection arrows (< and >) like OS/9 and UNIX. It has several other enhancements beyond the original FLEX features too. Both REXDOS and SK*DOS can read and write FLEX disk format. I found that though SK*DOS 68000 formats a disk with the GIMIX flavor format, it can read from or write to a disk formatted on a SWTPc machine with no problem. What you can't do is to format a disk on the 68000 machine under SK*DOS and put data on it, then read it on a SWTPc system. (Of course I am speaking of double density and or double sided disks).

By my personal test (a while ago) the 68000 16 MHz version is around ten times faster than the 2 MHz 6809. REXDOS has very fast floppy disk operations. Either REXDOS or SK*DOS will be disappointing with regard to hard disk accesses and data transfer rates when compared with a fast 386 or 486 clone system. Data transfer is not painfully slow, just not spectacular. Generally there are no 500K byte software packages for the 68000 systems. My PAT is about 33K of object code.

SWTPc 6800 Systems

The first computer that I bought was one of these. It squandered its memory map mercilessly. It allowed for RAM from $0000 to $7FFF. That is all of 32K. MiniFLEX used the $7000 to $7FFFF block. The processor board had a 128 byte static RAM that was used for the monitor stack at addresses $A000 to $A07F. I/O was decoded in the $8000-$9000 block -- very loosely decoded. Though only 32 bytes were actual I/O addresses, all of 8K bytes of the memory map were "wasted". Essentially the user had 28K of memory to use, from $0000 to $6FFF. The 128 byte static RAM (a Motorola 6810 if I remember correctly) wasted all of 4K in the $A000 range. $B000 to $DFFF were not used for anything, and a monitor ROM called McBUG was resident at $E000 to $E1FF. At the time MiniFLEX wasn't called MiniFLEX. The name was coined later after the larger FLEX2 came about. The original was called FLEX version 1.0.

MiniFLEX was a 4K operating system. It was tight. It had a "Utility command space" of 320 bytes inside of the FLEX area. That meant you could run small utilities from other programs and not write over them. Utilities larger than 320 bytes (and there were numerous ones) had to run in program memory space.

Very early SWTPc 6800 systems came with a "bit banger" serial interface called the MP-C board, essentially a parallel port that could output or input a bit at a time under program control (using a program loop for timing). The baud rates available were 110 and 300 (110 for teletype machines used as terminals) and it could operate with RS-232 levels or 20 Ma current loops with teletype equipment.

Later the 6850 was used. This original MP-C serial port required the program to do nothing but wait for a character, since it had to run its timing loop while waiting for each bit. The 6850 MP-S board had it's own baud rate clock input and it could input a character complete and then transfer it to its receive buffer. It could be reading the next character while you grabed the previous one. You only had to poll the "Receive Buffer Full" status often enough to insure that you read a character before the next one was completely received. The computer could do other things while waiting for the RBF signal. Outputting of characters is done similarly, polling the TBE status (Transmit Buffer Empty) and stuffing a new character into the port when the buffer was empty. Again there is a one character buffer.

I'm sorry but there is a lapse in my saved documentation. I can't find a schematic for the 6800 processor board. I believe it contained a baud rate generator consisting of a crystal and a 14411 baud rate divider that provided several baud rates to the buss. The MP-S serial card could be jumpered to one of the baud rate lines on the motherboard. Rates up to 9600 baud were provided.

Rounding out the available I/O was the MP-L parallel port board using the 6821 "Peripheral Interface Adaptor" usually called a PIA. The 6821 contains two complete 8 bit parallel ports, each having two handshake lines, one an input line and the other programmable as input or output. Generally the output handshake line was used as a "strobe" signal for a parallel printer, and the input only one was used either for "busy" or "acknowledge" back from the printer. When you sent it a character you either waited for busy to go away or for acknowledge to be received before sending another character.

6809 Systems

When the 6809 processor board came along, it required a newer motherboard or some modifications to the original one to move the I/O addresses up to $E000 and to tighten the decoding to a 2K block. I chose the modification route and installed a switch so I could move back and forth between the 6800 and 6809 boards.

There are 8 I/O slots across the back of the SWTPc computer. They are numbered port 0 to port 7. Port 0 is at the left rear as you face the front of the computer. That slot was usually used for a modem port at address $E000. The early 6809 computers used four addresses per slot, so the second slot was addressed at $E004. This was the slot for the terminal interface since SWTPc computers generally ran a serial terminal. Further slots are spaced four addresses apart at $E008, $E00C, $E010 etc to the last one at $E01C.

If you are unfamiliar with Hexadecimal arithmetic and Motorola's notation, please read the "Side Trip" section at the end of this. Our I/O memory address of $E000 is 8K below the top of memory. The ROM monitor in the 6809 systems is at address $F800. SWTPc called the 6809 ROM monitor program SBUG-E.

I am assuming (maybe wrongly) that most of you who are interested have a 6809 version. Later models had bigger I/O boards containing dual 6850 serial ports (MP-S2) or dual 6821 parallel ports (MP-L2). These models decode 16 addresses per I/O slot. The first is of course still at $E000. It has to be a dual serial port. The second port (of this first slot) is addressed at $E004 and is the serial terminal port at the same address as in earlier computers so the same operating system and ROM monitor can run them.

Perhaps I ought to explain why I/O slots require multiple addresses. The 6821 contains four registers. Suppose a parallel port were in a slot whose address is $E010. The parallel port would use four addresses for (in order) Port A data, Port A control, Port B data, and Port B control register. The four registers would be at consecutive addresses, the Port B control register being at $E013. The I/O buss across the back of the computer has the low order address lines available. These are connected to the RS (Register Select) inputs of the 6821 or 6850 chip. The serial port uses only two addresses, a data register and a control/status register. The later version of the dual parallel port, the MP-L2, used in the 16 address per port motherboard version, has a hardware buffer. The older version had no buffer and you could program bit by bit which ones were input and which output. The dual version has the buffer which can be programmed by writing to an address outside of the range of the 6821 registers, to make a whole 8 bit port either all input or all output.

Usually a well written printer driver in assembler would include an equate such as PORT EQU $E040. If you plugged a parallel port into port 7 you would simply change that equate to PORT EQU $E070 and reassemble the program. Symbolically in the Assembler program, Port A would be PORT. Port A control would be PORT+1, Port B would be PORT+2 etc. PORT would be what we would call the "Base address" of the parallel port chip.

FLEX leaves a small memory area available within the FLEX loading area ($C000 to $DFFF) for printer drivers. Since we can't possibly discuss everything in one article, we'll save that for a later column. Most FLEX utilities are written so they load and execute at address $C100. Some larger programs allow you to execute a system utility without exiting the program. Generally any utility program that loads at $C100 and fits between there and $C7FF can be run without hurting the presently running program. The mechanism for this is the fact that FLEX can be called as a subroutine. You set up the command line and point at it with the X register and then call FLEX. The command is executed and control returned to your program via an RTS (return from Subroutine).

Most large programs load at address $0000 and can use space up to $BFFF on a 6809 system. FLEX uses a small stack area within the FLEX loading addresses. Many small utilities just leave it where it is. Most large programs locate it at the high end of user memory, $BFFF or near there. The 6809, predecrements the stack pointer before it puts something on the stack, so you can actually use $C000 (the first address for FLEX) as the stack address because it will be decremented to $BFFF before anything is placed in memory.

FLEX was supplied with a good array of utility programs, all of them being "transient", that is, there were no memory resident utilities, not even a directory facility. Anything you wanted to do involved loading a utility from a disk, letting it do it's thing, and then essentially throwing it away by loading something else over it when you wanted to use another utility. Earlier, I mentioned that Miniflex had a "Utility command space" of 320 bytes for such utilities. 6809 FLEX uses $C100 to $C7FF or about 1700 bytes. There is a mechanism by which you can install one or a number of small utilities so they are memory resident.

Because FLEX doesn't read any disk information to memory there is no such action as "mounting" a disk or changing drives as such, (the B: A: selection of MS-DOS). One of the nicest features of FLEX in my opinion is the concept of a "System drive" and a "Working drive". You put the operating system, utilities, assembler, compilers etc. on one disk and call it the "system disk". Usually you put that disk into drive 0 and your "working disk" into drive 1. You set up your STARTUP file (just like AUTOEXEC.BAT and CONFIG.SYS all in one file) to make drive 0 the system drive and drive 1 the working drive. If you have other drives you can make any of them the working drive by using the ASN utility. ASN W=2 would make your third drive (drive 2) the working drive. ASN W=A will make ALL drives the working drive and FLEX will scan each drive in turn until it finds the file you've asked it to work on.

Commands default to the system drive and any filename to be used by the command defaults to the working drive. Including the drive number overrides the default. Command files have the extension .CMD. Most editors default to a file extension of .TXT for files to be edited. FLEX accepts the drive number in two different ways. 1.file.txt is valid but so is file.txt.1 as a file description. If you plan to go to SK*DOS (68000 version) eventually, don't use the trailing drive number. SK*DOS doesn't really support it largely because of the multiple directory scheme that it uses.

The command "ASMB FILE +LSY" will invoke 0.ASMB.CMD on 1.FILE.TXT (if you have set your working drive to 1) and produce an output file FILE.BIN on the working drive with no Listing file (+L), no Symbol table (+S), and it will delete any previous FILE.BIN (+Y for Yes, delete the old file). It is not quite like having multiple directory facilities, but then who needs multiple directories and treelike paths on an 85K capacity floppy disk? If you want multiple directories you simply change working disks. I had a disk labeled BASIC Programs, one labeled Assembler, one labeled Correspondence, etc.

FLEX, though originally supplied directly by SWTPc to their customers who bought disk drive hardware, was written by programmers at a company called Technical Systems Consultantants, then located in West Lafayette Indiana (home of Purdue University). I'll refer to them as TSC below.

In addition to the utilities supplied with FLEX, TSC had a utility package that contained some memory test utilities and such niceties as UNDELETE, RAWCOPY, and others. There is a particularly useful utility called EXAMINE that lets you edit individual hex bytes right on the disk. You can look at any track/sector combination and edit any byte on any track and sector. There were several public domain "DISKEDIT" utilities available, some better than EXAMINE for that purpose. Individuals and user groups contributed better versions of many of the utilities from time to time.

That brings up an interesting memory. One time in a column in '68' Micro Journal I was discussing that we hobbyists ought to get past writing utilities that had already been written and go on to do something useful with our computers. I happened to say that if I saw yet another DIRectory utility I would probably get sick. Before the column was published, I received, guess what? Right; yet another DIR utility. It was sent by a reader who had spent a lot of time on it, and had made it do some nice things the original wouldn't do. That is, it had extra options. Of course I received a very nasty letter from the author saying he would never bother to read my column again. I wrote and explained the coincidence, but it was a couple of years before I heard from him again, saying that he had put my letter aside and had intended to answer it for a long time. He understood the coincidence and all was OK.

I had disassembled about half of FLEX over a long period and with great difficulty. I had concentrated on the command processor and not the disk drivers. Someone else had disassembled the disk drivers and not the command section, and we traded quietly. The folks at TSC said they had seen nice source listings to all of FLEX that were far better formatted and commented than the original source code! They didn't object to anyone disassembling FLEX, but would have been very upset at anyone making the source code public in some way. This was before bulletin boards, or perhaps that might have been done.

Programming the 6821

This is intended to be introductory, to give you a feel for how peripheral devices are programmed. Those who are familiar with Intel processors will find a couple of things that are different here. First of all, the 6809 has no separate I/O buss. The peripheral devices reside on the address and data buss, and they are accessed identically to other memory locations. Secondly, a 16 bit value in the Motorola processors is represented such that the high order byte is at a lower memory address.

The 6821 is reset on power up. All bits are initially programmed as inputs. To program them as outputs you must access the data direction register via the control register. Let's talk about the "A" side of the PIA, the one at the lowest memory address. The "B" side is essentially identical except that it's registers reside at the next two higher memory addresses above the A side. The Data A Register is at the base address, the address of the I/O slot. We'll call it BASE. The Control register is at address BASE+1. Bit 2 of the control register controls whether we access the data register at the base address or the data direction register at that address. (Motorola numbers bits such that the low order bit is b0 and the high order one of a byte is b7. The number corresponds to the power of 2 represented by that bit so bit 2 on by itself would be binary 00000100 or HEX 04). We clear the control register:

```
CLRA                ALL BITS 0 TO ACCESS THE
                    DATA DIRECTION  REGISTER
STA     BASE+1
STA     BASE        SET ALL INPUTS
```

If we want to make the port all outputs we write 1's.
Instead of the above we write:

```
CLRA                ALL BITS ZERO
STA     BASE+1
LDA     #$FF        WE COULD AS WELL USE
                     COM A TO SET ALL BITS TO 1
STA     BASE        ALL OUTPUTS
```

Then we must change the access from the data direction register to the data register by setting bit 2 of the control register.

```
LDA     #$04
STA     BASE+1
```

The two handshake lines are CA1 and CA2. They may be used as inputs. If CA1 is triggered, the control register bit 7 is set high. If CA2 is set up as an input and is triggered, bit 6 is set high. CA2 can be used as an output by your program. Essentially if you write $3C to the control register CA2 will go high and when you write $34 to the control register, CA2 will go low. That is, it follows bit 3 of the control register when bits 4 and 5 are high. Remember that bit 2 must remain high or we will access the data direction register rather than the data register.

Most printer drivers put an ASCII character into the data register which has been initialized as all outputs. Then they toggle the CA2 output by writing $3C and then $34 to the control register. Then they wait for an acknowledge on CA1 as seen by looking at bit 7 of the control register. The program can poll (i.e. read the control register repeatedly in a loop) until the bit 7 goes high as a signal that another character can be sent.

In addition to being able to read the control register to see the status of the printer handshake, it is possible to activate an interrupt when the signal is detected. Doing so is just a matter of the initialization of the control register. Since this is going to be rather long, we'll defer a discussion of interrupt driven I/O until later. None of the software supplied by SWTPc or TSC used interrupts.

If you have one of the newer MP-L2 interface boards you have an additional chore. As mentioned above, it seems SWTPc added a bidirectional buffer to the parallel port and the handshake lines. There is a "latch" at the address $0E above the port base address. That is, if the base address is $E010, the latch is at $E01E. Actually it is also redundantly addressed at $E01F. The latch is used as follows:

bit	function	
0	0 for PIA A side input	1 for Output
1	0 for PIA B side input	1 for Output
2	0 for CA2 input	1 for Output
3	0 for CB2 input	1 for Output

Note that you have to set the Data Direction Register AND the proper Latch bit to the same state. With the MP-L2 you can't have one parallel port with both inputs and outputs. The latch bits 0 and 1 set the direction for the entire 8 bits of the port. The other bits control four output signals that are set high or low with the corresponding bits. B4 controls OUT1 on the A side, B3 controls Out2 on the A side, Bits 1 and 0 control OUT1 and OUT2 respectively on the B side. The upper connector is the A side, the lower the B side. Pins 24 and 25 of each are the OUT1 and OUT2 signals respectively. If this is not sufficient information and you don't have an MP-L2 manual, get in touch and I'll send you a copy.

If you have a computer you hopefully have some of the original documentation. I'll send the data sheets (copies) for the 6820 and 6850 along with this. See the accompanying reproductions of them. Hopefully with what I have just presented you will be able to work through the 6850 programming, which is simpler. Note that the SWTPc hardware (the 14411) generates a 16X clock for the 6850, so you program it for that. I generally program a 6850 just one way. Unfortunately (just to make it harder to remember) the BASE address in the case of the 6850 is the control register. I initialize the serial port by writing $03 to the control register, then $15 which specifies 8 bits, 1 stop bit, no parity, clock divided by 16. Again, the data sheet copies will help. Rather than go into another whole article here, I'll offer to help anyone who writes me a note describing the difficulty. Please wait until after the data sheet information has been published.

There is a caveat with the 6850 as used in the newer MP-S2 boards. The CTS signal is brought out to pin 20 on the DB-25 connector. It is not pulled up to +12 volts, so the CTS is not asserted unless you have connected it to the RTS of your terminal or other device, or, jumpered it (pin 20) to pin 4 of the connector that plugs onto the MP-S2. The older boards had both the CTS and the DCD (Clear To Send and Data Carrier Detected) pins on the 6850 (pins 23 and 24) grounded. The older serial ports therefore not only don't need the CTS signal, they can't use it.

If I've gotten you confused let me say that a high RS-232 level on CTS (i.e. +12 volts) is converted through the Motorola 1489 RS-232 receiver chip to a low TTL level to be applied to the 6850. The input on the 6850 is an "active low" input so grounding it effectively turns it ON. SWTPc probably added the CTS input because their terminals (the CT-1024 and the later CT-82) couldn't run at 19.2 KBaud without the handshake. If the terminal couldn't keep up it would turn off the CTS and the computer would have to wait for it to catch up.

For reasons I don't quite understand it seems that SWTPc brought the CTS connection out to the DCD pin (20). But then if you've been through a few RS-232 interconnect projects you realize that "standard" is pretty elusive.

Lastly, I note that Bill Kibler published a list of software suppliers for the 6809 and 68000 machines and mentioned "Spray Software" as a source for Whimsical. Whimsical is a compiler for SK*DOS 68000 that is very much like Pascal, but it has numerous features that make it better for accessing hardware. You can declare a variable AT an address and use it to access an I/O port, for example. It is a single pass compiler and it can handle pre-compiled modules.

Being single pass it is fast, but every variable and procedure must be defined before it is used. It does allow "forward" declarations as does Pascal.

Whimsical has the BOOLEAN variable type of Pascal, but the convenience of pointers like those in "C", though the syntax is different, (read that "less confusing"). It has a nice "write" facility that will nicely format output regardless of data type for debugging statements. It is a sort of an automatic printf() more like the "cout" facility of C++. The variable type determines the format of the output. It is really handy for printing variable's values to the screen. I don't know what John Spray is getting for the compiler presently. My best guess is $50 U.S. His address is:

Mr. John Spray
4 Dianthus Place
Browns Bay
Auckland 1310
New Zealand

If by chance some of you are interested enough to write me a note, I'll do a comprehensive review of Whimsical in a future column.

Other possible future topics:

6809 Assembler programming
Complete I/O drivers in Assembler
I/O drivers in higher level languages
Reviews of software
(I'd like to wait and review things that are available)
Continued discussion of hardware
Initialization and serial in/out on an 8250 (on a PC clone)

I'll consider a note from you as a vote for or against one or more of the above if you will tell me what interests you most. If you have an interest in a topic not mentioned above, suggest it to me.

My Address: Ronald W. Anderson
3540 Sturbridge Ct.
Ann Arbor, MI 48105

Please write rather than call me. I can read letters at MY convenience. I must answer calls at YOUR convenience. I hope you understand. If you feel that you MUST call, please call my office phone after 6 PM Eastern Time and leave me a voice mail message. The number is (313) 769-6640.

A Little Rambling

One day several months ago I was driving to the computer store to pick up some 1 megabyte SIMM memory modules, four of them, to be exact, at $32 each. I started to think about how much progress there has been in so few years (at least it seems like few to me). In 1958 and 1959 I worked on a project (I am an electrical engineer by education) that used a 256 BYTE ferrite core memory, 256 by 8 or 2K bits. At the time, the target was a dollar a bit cost for that memory.

In 1977 at the start of the "microcomputer revolution", 1K by 1 memory chips were about $3 each or about .003 dollars a bit. Here we had better shift to 0.3 cents a bit. Later I was able to buy a 16K by 8 board for about $128 or about 0.1 cents a bit. A while later we were buying 64K by 8 static CMOS RAM boards for $128 or so, about 0.013 cents a bit. As I drove to the computer store, I realized that I was going to buy 8 megabits of memory for $32 or about 0.004 cents a bit. That, my friends is a cost reduction ratio of 25,000. If automobiles had become less costly by the same ratio, you could now buy a Chevrolet for less then ten cents. (In 1957 I bought a Chevrolet for $2300). Now I grumble because due to the "shortage" of plastic, (a plastic plant in Japan burned down) RAM prices have gone up to $49 per 1 Mbyte SIMM. Actually they went from $39 to $59 each a few days after the company bought a system with 8 megs of them. Now they are back to the $49 price level. I not only grumble, I am holding off buying any more until they come down to the less than $40 range again!

Hexadecimal Numbers

If you are familiar with Hexadecimal numbers and Motorola's symbol conventions you can skip this. In Motorola's assemblers and many of the compilers that run under FLEX and generate 6800 and 6809 code the dollar sign ($) preceeds a hexadecimal value. Each Hexadecimal digit stands for four binary digits (bits) as follows:

Binary	Hex.	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Hexadecimal can be seen to be shorthand for four bits of binary value which can have the values of 0 or 1 only. Binary is easy to understand. Think of the rightmost digit as representing 1, the next digit as representing 2, the next 4, the next 8 etc. The value 1010 represents 1x8 + 0x4 + 1x2 + 0x1 or 8 + 2 or 10 decimal. What does the next left binary digit stand for? If you said 16 you are correct. What is $64 in decimal? First spell it out in binary: 0110 0100. If you do as we did above you get 4 + 32 + 64 = 100 decimal. Note that decimal runs out of single digit representations at 9. Hexadecimal simply borrows the first six letters of the alphabet to represent decimal values 10 through 15.

If each binary digit as we go from right to left is worth twice the previous, each hexadecimal digit must be worth 16 times as much as the digit to it's right. Looking at $64 again, the left digit is worth 6 times 16 or 96, and the right digit 4 is worth 4 times 1, so we have 96 + 4 = 100. After a little practice it is easier to work with shorter numbers in hexadecimal than with long ones in binary notation. Just as in decimal we have 1000's, 100's, 10's and 1's, in Hex. we have 4096's, 256's, 16's and 1's.

One K of memory is 1024 bytes which translates to $400 in Hexadecimal. $1000 represents 4096 bytes or 4K. A handy rule for estimating the precision of a decimal number based on the number of bits in the binary code is that ten binary bits are equivalent to three decimal digits. For example 64 bit floating point numbers might use ten bits for the binary exponent and 54 for the mantissa. The decimal resolution will then be about 16 digits.

Special Feature

All Users

The Historic ZX-81

Sinclair ZX-81

by Bill Kibler

Over the past year, I have received many letters that commented on how the ZX81 and it's relatives were responsible for the reader becoming computer literate. That literacy covered the full range of user skills. Some only learned what computers could do, whiles other became real hardware hackers.

I remember being at a friends house and playing with one of these units. We didn't play very far that night, because it was a used system, and the contacts for the memory expansion interface kept failing out about halfway through entering a program.

Although my first experiences were not positive, many of our reader's were. As I read these letters (and you can too, later), I was struck at how many people went beyond my first experiences and adapted and developed rather complex projects that worked well.

The important aspect of these systems is that they were learning platforms. They could be bought cheaply, used with minimum items (typically the unit and a power transformer, with connection to your TV set). This meant that you focused on the operations of the device and not how many peripherals you could hang off of it (although some took this as a separate challenge in itself.)

The biggest shocker arrived one day in the mail. W.D. English of Orange CA, decided to send me his collection of ZX81 books. This rather large box arrived with 27 books inside. I list these later so you can see what was available. The quantity is amazing when you consider how limited these machines were.

One explanation for the quantity of books is that the ZX81 was the first mass marketed computer. Timex, a maker of watches, wanted to diversify into the new and hot field of computers. This English made system they supposed was just the vehicle. The consideration that sellers would also need to support the user had never occurred to them until after they started selling them.

A good review of the Sinclair ZX81 history can be had by getting Stan Veit's History of the Personal Computer (a recent book). Needless to say many of these units were sold, especially when Timex started dumping them for practically nothing. Even with no hope for support many bought their first computer for under $50 and were on the way to becoming computer users.

Lets hear now from some of those who got their start by using ZX-81's.

Dear Bill,

I have just finished reading issue number 63. Articles about the ZX81 would be great. By the way, the ZX81 was a kit, but TIMEX also sold them under the name TIMEX/SINCLAIR 1000, available anywhere TIMEX watches were sold. The T/S 1000 came with 2K of memory and the ZX81 came with 1K! This was enough to get started in BASIC and later you could upgrade to 16K. This machine was my first computer. I bought it as surplus for $20 from a catalog. My parents were not very happy at my desire to own a computer. But after using an Apple II+ in high school I just couldn't think of anything else. I would still be using that computer every day but I couldn't find a disk drive system for it. Also, several of my machines need the ULA chip, where can I buy them? [can anybody help Roger? Bill K.].

Roger Olson, 2304 West 4Th, North Platte NE 69101.

Dear Bill.

My second computer was a ZX81 - the first was a Lambda, a ZX81 clone with a much improved keyboard. This computer was also sold under the name of Power in all big toy stores. Almost all BASIC-routines were exactly identical to the Sinclair ZX81 routines. Just the addresses were different. I learned Z80 code by comparing the BASIC-ROMs by hand, a very instructive exercise. I would use the information so I could add new BASIC words and I ended up with a very useful extended BASIC on the Lambda computer.

I have ASZMIC and FORTH for ZX81 and ASZMIC for the Lambda. Any interest? Just drop me a line and an EPROMs. ASZMIC fits in a 2732 and FORTH in a 2764. [Gorm; can you read the proms and put them in Intel hex format on a PCDOS disk and send them to me for distribution to my readers who want them?? Bill K.] I look forward to reading about the emulators. Please write how we in Europe can get them.

Gorm Helt-Hansen, Domusvej 4, DK-2680 Solroad strand, Denmark.

Hi Bill,

The Sinclair ZX81 was my first computer. I bought the kit for $99 which was a very low price at the time. It had 1K bytes of RAM. The operating system and BASIC was in 8K byte ROM.

This was my first project and I must have triple checked everything. I finally got the computer assembled and turned on. Seeing the prompt on the screen for the first time was a thrill I still remember.

On the PCB, IC4 actually has 3 locations; IC4, IC4A and IC4B. IC4 has 28 pins and inside that outline is IC4A with 18 pins. A separate location is marked IC4B with 18 pins. My computer had no socket at IC4B and a 28 pin socket at IC4 with a 4118 RAM installed. (1K bytes). Later machines had as much as 2K bytes of memory.

The system bus could be accessed at the rear of the housing and I built many devices for it, including memory expansion, ROM readers and controllers. I even modified a B&W TV so that I could directly plug the ZX81 into the video drive amplifier for a clear video display. The worst part was the membrane keyboard and I finally got a key click replacement.

Your center spread on the XEROX-820-II is very welcome. I have one that I built and still use once in a while.

Sincerely, Herb Goldseger, Oakland, CA.

Dear Mr. Kibler

In a recent issue of The Computer Journal there was mention of a need for a Sinclair ZX-81 schematic; enclosed is a copy of same. I also saw a request for information on repairing the keyboard. I wrote an article awhile back on repairing the flexible circuit connection between the keyboard and the motherboard, for which I could not find a publisher. Perhaps your readers would have an interest in it, as it includes photographs to document the procedure.

I have also written article describing an elementary stepper motor circuit which illustrates the basic principles of operation. If these articles are of interest I can send you copies to evaluate. [please send them to me, Bill K.].

Regards Tom Poulin, Southbury, CT 06488.

Dear Mr. Kibler,

In issue #61 you mentioned in the 'READER to READER' column that you'd like a schematic of a ZX81 for publication. I don't know how many of your readers assembled their own ZX81's, but I did and therefore I have the schematic. You'll find it enclosed in this letter. The assembly guide is in German but I guess you'll be able to extract the essence.

Unfortunately I managed to sell my ZX81 when the 'ZX Spectrum' came, so I don't have it any more, but back then I thought I made a good deal.

BTW (By The Way), did you know that some of the people behind the ZX81 made their own company and released a FORTH machine named Jupiter ACE?? It was based on the same ideas as the ZX81, but it had a better keyboard (a bit like the one from the ZX Spectrum). It was incredible fast for it's time and price. Maybe some of your British readers could dig one out. [A USA reader did just that and information on Jupiter will be in #66, Bill K.].

Thank's for a great journal. Yours sincerely Michael Dantzer-Sorensen, Denmark.

Hello Bill

Here are the schematics and assembly instructions for the ZX81/TS1000 and 16K RAM pack. The ZX drawings are copies of the docs supplied with the kit, the ram schematic is from SYNTAX Quarterly, a TS1000 magazine. I'm telling you this in case you might have a legal problem in republishing the material. The RAM schematic looks hand-drawn and Britishly so SQ may have lifted it from someplace else themselves.

One warning for any unfamiliar person using the ZX81 schematic: the PC boards were manufactured to accommodate either a single 4118 RAM chip (1K x 8 bits) or a pair of 2114 (1K x 4 bits) RAMs. Unless you read the fine print on the schematic, it looks like all three are installed. I think that Sinclair did this so that they could use the same board with whatever RAMs they could buy back in 1981. Whichever way, just 1K (and lets not go into the addressing scheme, its close to lunchtime!)

Ken Smith, Sunnyvale, CA.

To: B.KIBLER

Sub: JEEZ, BILL

Taking my money, forgetting my name, and then asking me for a schematic? Hey, Bill, you should work for the government, you've got the attitude! Now that I've properly insulted you, let me make amends by sending you the schematic and assembly instructions for a ZX-81 (aka TS-1000) via snail mail. The Sinclair was a remarkable machine for its time and price, but couldn't compete with machines like the Commodore VIC's and 64's; which offered color, plug in cartridges, and a real keyboard for a few dollars more. (I think that there's a lesson to be learned that one sure way to make a product flop is to go cheap on the keyboard, for examples see IBM PCjr, Poquet PC, and the current generation of digital address books)

I built my ZX-81 from a kit, it cost $150 along with an extra 16K memory cartridge. I later added a Suntronics keyboard and Computer Continuum expansion board, and Timex printer. A lot of stuff for the little guy! Sinclair later had a 68008 based machine out but I don't know if it was seriously marketed here in the US, this was after their deal with Timex. I see you have a schematic for the Xerox 820 in your current issue; I have one of those boards, too, but I had too many other projects to get far with it, but I do have the BIOS ROMS. So much for the ancient history. I've got a half written power supply article I should finish and send you; maybe TCJ doesn't pay for auricles, but a lot of mags don't pay much more! Keep up the good work!

Ken Smyth, K.SMYTH1 on genie.

PS Your reader who had keyboard problems should be pleased to see that the keyboard is a very simple matrix, and could be replaced by a normal "dumb" (no processor inside) keyboard without much difficulty other than checking and maybe changing the matrix in a few spots. Surplus stores usually have piles of these at giveaway prices.

PSPS In response to your Jupiter question, it sounds vaguely familiar. Does anyone have a "ZX.ASMIC" rom? This was a replacement rom that turned the ZX81 into an assembly development station for Z80's. It was made by a UK company called Comprocsys. They also had a teletext adapter and a external card for the little beast. The ASZMIC rom (now spelling is correct!) is what I'd most like to get.

Dear Bill

In Issue 61 you briefly mentioned that you were interested in finding a copy of the schematic for the Sinclair ZX81. Consider it found; here it is.

Like many others, I cut my teeth on the ZX81 (built from a kit). I still have three of them around (picked up at swap meets) and lots of manuals, magazines and books.

I used to belong to the Vancover Sinclair Users Group where we had a very active hardware group. We designed and built a 32K battery backed RAM disk, speech board, I/O board and other minor memory boards. Our software friends wrote High-RES graphic software (256 by 192), multi-tasking software and many other things that others would have sworn was impossible.

If you or your readers have any questions, I can try and answer them. I haven't used it in years but I can still find most of the documentation.

Eric Sakara, 200 Killarney Ave. Winnipeg, MB Canada R3T 3B5.

Dear Mr. Kibler:

I just received my copy of issue 61 of The Computer Journal, and saw your response to Ken about the availability of schematics for the venerable Timex Sinclair ZX-81. I am going to go one step farther, and am sending you a copy of the schematic diagrams that I have for both the ZX-81 and its predecessor, the Sinclair ZX-80.

Most interestingly, last week I was logged into the White Sands Public Domain software pool, SIMTEL20, and in the messy-dos directory under emulators and I find the index has an emulator for these machines. Maybe someone will download this emulator and give us a report on it. I can't because it will not run under either of my CP/M systems.

Sincerely, Paul V. Pullen, Baltimore, MD.

Dear Mr. Kibler:

The spurt of interest in Sir Clive's machine exhibited in your latest issue has convinced me to send you, by separate mail, all the various books about the Timex Sinclair that I have accumulated over the years. Note the comments in some books about how it is good for the soul to be limited to 2Kbytes (or 1Kbytes) of memory.

Yours truly, Dave English, Orange, CA.

Thanks Dave and here is an annotated list of those books, sorted by size (I stacked them on the floor using large ones for the base).

MASTERING YOUR TIMEX SINCLAIR 1000 PERSONAL COMPUTER, by Tim Hartnell and Dilwyn Jones. A pocket book by Batam Books and printed in March of 1983. It has 18 chapters plus an index. There is a very brief introduction and then each chapter explains a Basic function or concept. There are many examples to try and the book is well suited for beginners.

TIME LOST, by Joseph C.Giarratano, Arian Keith Andrews, SR., Kris Austen Andrews. A pocket book published by Que in 1983. This is one of a series of books that have a comic book style story with characters that encounter action based adventures. The idea is to program in a arcade type of game when the story encounters the peril. The six games are short Basic programs listed in the back of the book. Each game had a short explanation of how it worked and was used. It appears to be a nice way to tie playing games with reading stories and all the time learning how computers work. Similar books for other machines were also produced by Que Books.

Continued after Centerfold

TCJ Center Fold

Special Feature

All Users

Sinclair ZX-81

9. POWER SUPPLY

If you wish to use your own power supply with the ZX81, it should conform to these specifications:
D.C. only – positive to the tip of the 3.5mm jack plug. Need not be regulated, but should be well smoothed.
Voltage – between 12 volts maximum and about 8 volts minimum (depending on smoothing) when on load.
Current – not less than 600mA, or 1.2A of the printer is to work from the same supply.

(c) IC2 and IC4 have two different sized boxes printed on the board: use only the holes corresponding to the smaller box.
(d) As previously mentioned, IC4 may be in either one or two packages. Only the appropriate IC sockets will be supplied, so make absolutely sure you know which version you have got before proceeding.

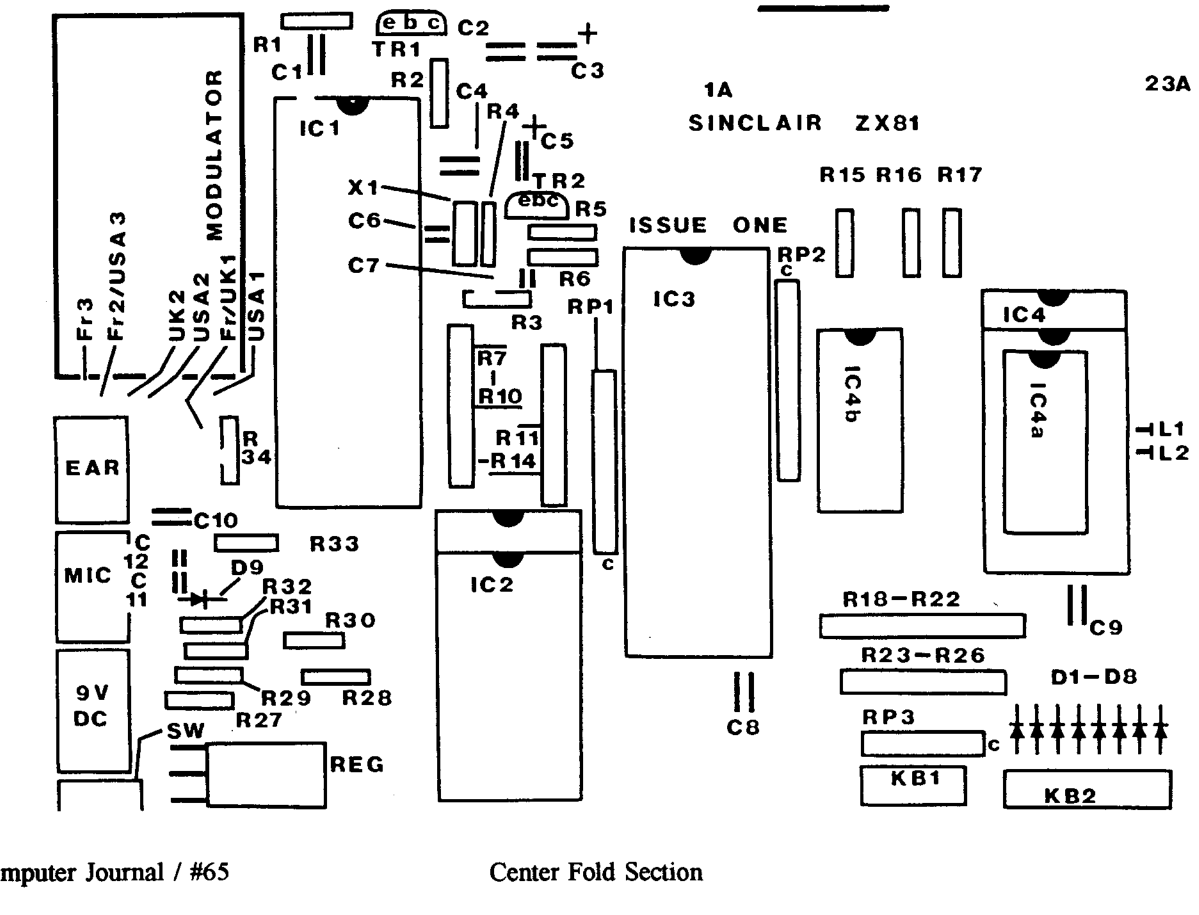
IMPORTANT:– If you have the 24 pin 4118 in your kit, a short wire link should be inserted in the holes at position L1. Use a component lead off-cut for this. DO NOT do this if you have two 18 pin 2114s, and do not put anything in position L2.
(e) When mounting the regulator, do not bend its leads too close to the plastic. Bolt it down firmly with its heatsink *before* soldering.
(f) Put the modulator's wires through the holes marked "Fr/UK1" and "UK2". Put each lead through the hole it is nearest to: do not cross them over. Do not try to bend the thick pins on the modulator: hold it in place by hand whilst soldering. The black card trim is a push fit over the aerial socket.
(g) The I.C.s will have their pins splayed out slightly and you may need to push them inwards slightly, e.g. by pressing against a flat surface, before they will fit the sockets. Make sure that each pin has in fact gone into its respective socket and that none are curled up under the I.C.

9. Netzteil

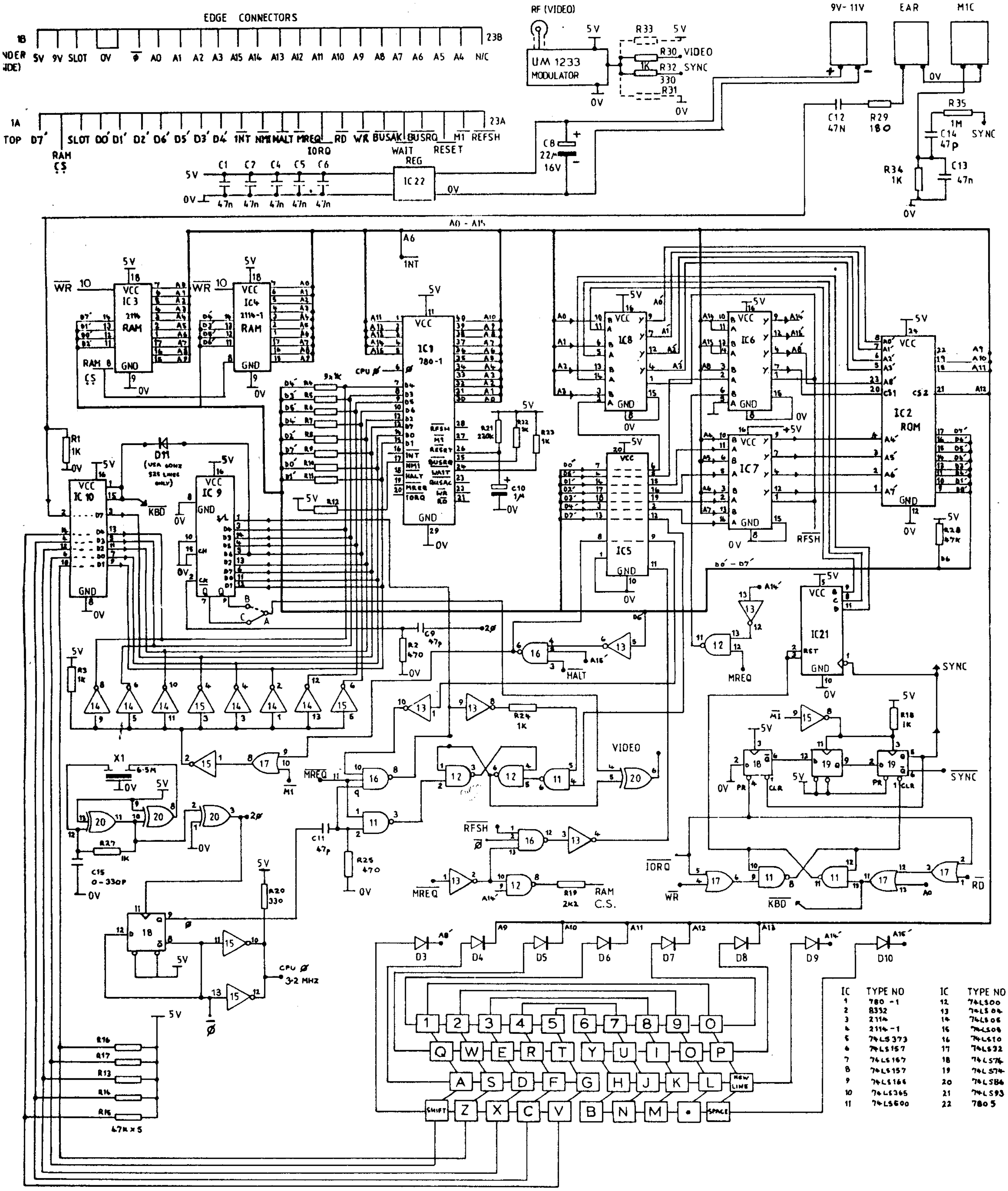
Falls Sie mit dem ZX 81 Ihr eigenes Netzteil einsetzen wollen, sollte es den folgenden Daten entsprechen: Nur Gleichstrom – positiver Pol an der Spitze des 3,5 mm-Steckers. Muß nicht geregelt sein, ausreichende Glättung ist jedoch erforderlich. Die Spannung sollte bei 12 Volt maximal und 8 Volt minimal (abhängig von der Glattung) liegen. Stromstärke nicht unter 600 mA oder bei Druckerbetrieb nicht unter 1,2 Ampère.

(c) Zu IC 2 und IC 4 gibt es je zwei unterschiedlich große Umrißzeichnungen auf der Platine – benutzen Sie nur die der kleineren Zeichnung zugeordneten Löcher.
(d) Wie bereits erwähnt, besteht IC 4 entweder aus einer oder zwei Komponenten. In jedem Fall werden nur die passenden IC-Sockel mitgeliefert – vergewissern Sie sich also vor Einbau genau, welche der beiden Versionen bei Ihnen vorliegt.
WICHTIG: Falls Sie mit dem 24füßigen IC 4118 arbeiten, muß eine kurze Kabelverbindung zwischen den Löchern bei Position L 1 hergestellt werden (mit einem abgetrennten Stück Anschlußdraht). Es darf allerdings keine Verbindung hergestellt werden, wenn Ihr Bausatz mit zwei 18füßigen IC 2114 bestückt ist. In keinem Fall darf etwas in Position L 2 angebracht werden.
(e) Bei Einbau des Reglers sollten Sie darauf achten, daß dessen Anschlüsse nicht zu dicht am Plastikgehäuse gebogen werden. Vor dem Festlöten muß er erst zusammen mit dem Kühlblech festgeschraubt werden.
(f) Führen Sie die Anschlüsse des Modulators durch die mit Fr/UK 1 und UK 2 gekennzeichneten Löcher – jeden Draht in das in seiner unmittelbaren Nähe befindliche Loch (Drähte hierbei nicht überkreuzen). Versuchen Sie nicht, die starken Drähte des Modulators zu biegen; fixieren Sie ihn beim Löten mit der Hand. Der viereckige, gelochte Karton dient als Aufsatz über den Antennenanschluß.
(g) Bei den IC's kann es unter Umständen notwendig sein, deren Anschlußstifte leicht, für die Sockel passend, zu biegen (durch Pressen gegen eine harte, glatte Fläche). Gehen Sie sicher, daß sich wirklich jeder Stift in der entsprechenden Sockelaufnahme befindet und sich keiner unter das IC gebogen hat.

FIG 6



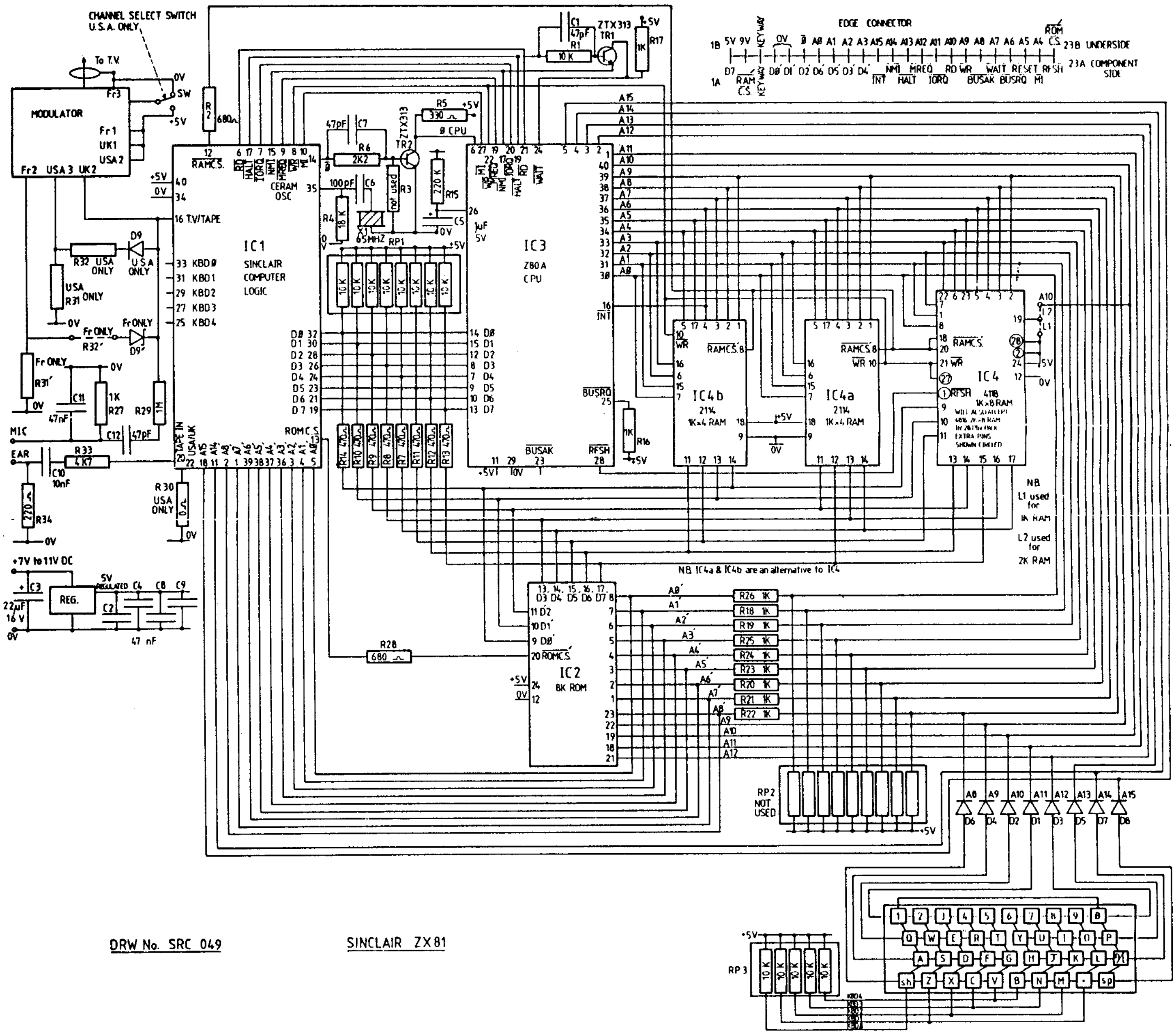
ZX80 CIRCUIT DIAGRAM



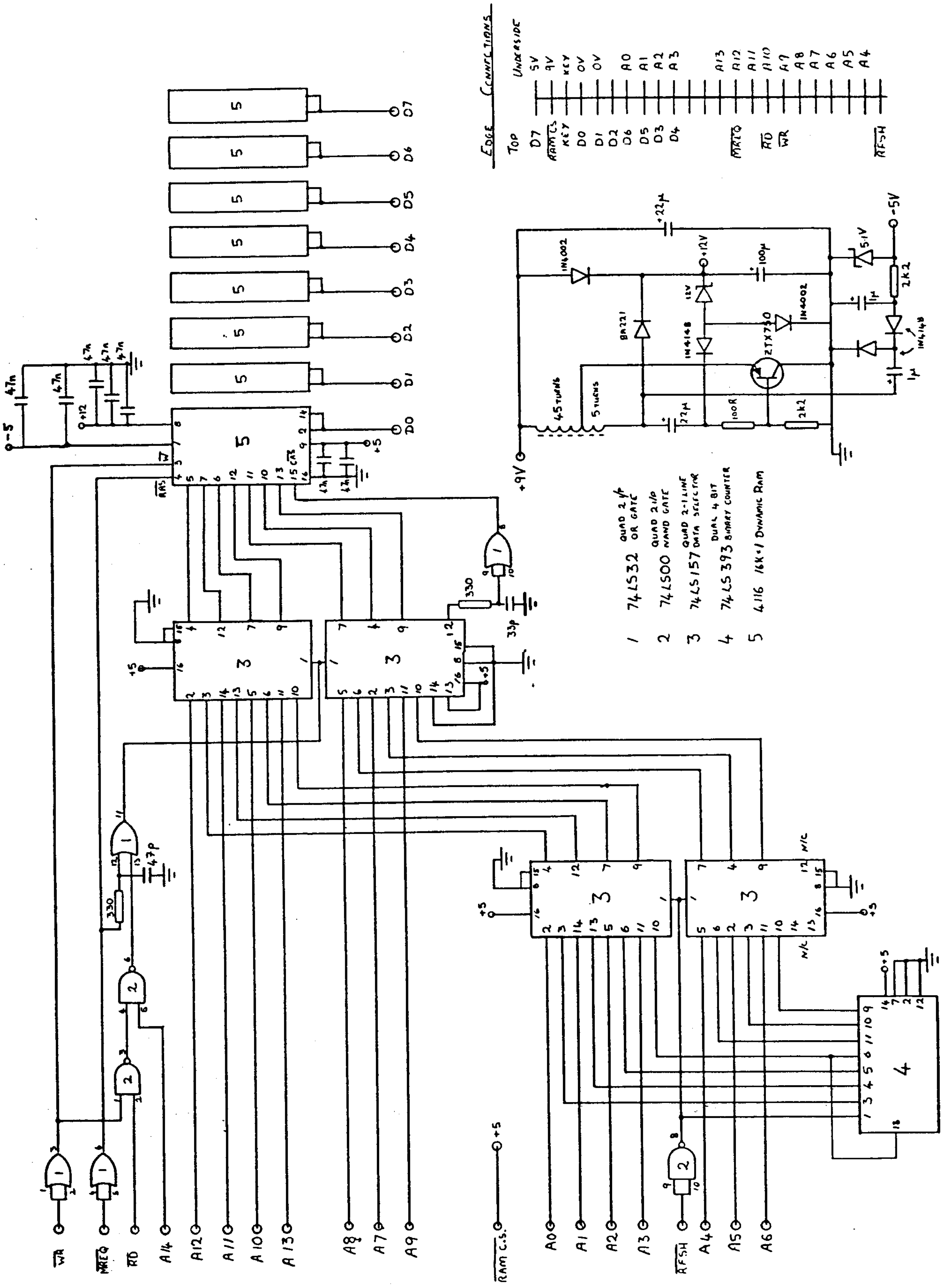
IC	TYPE NO	IC	TYPE NO
1	780 -1	12	74LS00
2	8332	13	74LS04
3	2114	14	74LS06
4	2114 -1	15	74LS08
5	74LS373	16	74LS10
6	74LS157	17	74LS32
7	74LS167	18	74LS74
8	74LS157	19	74LS74
9	74LS166	20	74LSB6
10	74LS365	21	74LS93
11	74LS600	22	7805

ZX80 and ZX81 Schematic

Presented here are the schematics for ZX80 and ZX81. The reason for also presenting the ZX80 schematic, is that it shows the interface circuits that were replaced by IC1.



16K RAM Pack Schematic



Sinclair ZX-81, Continued from before Centerfold:

THE TIMEX PERSONAL COMPUTER MADE SIMPLE, by Joe Camplbell, Jonathan D. Siminoff, and Jean Yates. A pocket book from Signet in 1982 that covers all the basics of using and programming the computer, again in Basic. The getting started section is nice in that it has both pictures of the connections and a layout of the keyboard (which helps understand the special keys). They use small graphics to help give the book a better beginners appeal. The text appears to cover all topics with a special appendix section that included a listing of people and organizations that supported the computer in 1982.

51 GAME PROGRAMS FOR THE TIMEX SINCLAIR 1000 AND 1500, by Tim Hartnell. This pocket book by Signet is just what the title says. 51 games are listed, some very short and simple, other long and complex. Only explanation is how to run the game not how the program works.

THE INS AND OUTS OF THE TIMEX TS1000 AND ZX81, By Don Thomasson. This Melbourne House book (of England and Australia, 1983 vintage) is excellent for hardware people. The first text is not text but the full schematic. There are only two section, INTERNALS and EXTERNALS. The internals section describes how the machine was designed and works. The external section shows how to added many new devices and circuits.The finals example is how to use the system to control a model railroad layout.

TIMEX USER MANUAL for the Sinclair 1000. This of course is the manual that came with the unit. As far as manuals goes this one is not too bad. There seems to be plenty of good examples and explanations of the Basic functions and options. A minor explanation of machine code and other options is made, but Basic is what the machine does best and the manual reinforces that aspect.

UNDERSTANDING YOUR ZX81 ROM, by Dr. Ian Logan from Melbourne House Publishing. Done is 1981, the title is very much not what the book describes. The book mainly deals with machine code and how to do it using the Basic POKE function. About all you use the ROM for is calling some of the functions from your machine code routines poked into memory. For those wondering about what machine code is (the language of the CPU or Z80 in this case) the book appears to provide a good explanation and some programs to try.

MAKING THE MOST OF YOUR ZX81, by Tim Hartnell from Reston Publishing (Printice-Hall). A rather brief explanation of Basic and then the book provides a large number programs and games, all without much explanation as to how they were programmed or why.

THE ZX81 POCKET BOOK, by Trevor Toms from Reston publishing in 1981. Like the previous book we have mostly examples of Basic programs, but unlike the previous, we get detailed explanations of how these programs work and even a few " Hints 'N' Tips".

TIMEX SINCLAIR 1000/ZX81 USER'S HANDBOOK, By Trevor J. Terrell and Robert J. Simpson from Howard W. Sams & Co., in 1983. This appears to be a three part guide to using the ZX81. The first part being a beginners introduction to the machine and Basic. Next is a more advanced Basic programs section. And lastly is the hardware and assembly language sections. The last sections shows some interfacing schematics and provides the Z80 instruction set in a way that probably is not very helpful.

THE GATEWAY GUIDE TO THE ZX81 AND ZX80, by Mark Charlton, from Creative Computing Press (1981). A rather poor book of program samples without explanation. They put pictures in between sections of code to make you think they explain something (they do not). The listings are from a very low quality dot matrix printer and extremely hard to read.

USING & PROGRAMMING THE ZX81/TS1000, by Albert N. Sickler, from TAB books in 1981. A fairly straight forward explanation of Basic and how to use the ZX81. Mostly explanation of Basic with the programs reserved for the appendix. They try to explain everything you might need to know to use the machine. Much better than the previous few books.

THE COMPLETE TIMEX TS1000/SINCLAIR ZX81 ROM DISASSEMBLY, by Dr. Ian Logan and Dr. Frank O'Hare, a Melbourne House publication. This is the ROM code disassembled so to speak. It is an important book for those wanting to write directly to the built in ROMs. Using the book however will take some time to understand. The format is not one I would recommend, but all the code and what it does is listed here. Some sections provide considerable explanation, but it is done in the sequence of disassembly and thus the order is somewhat random. A must have for the serious hacker!

ZX-81/TIMEX, PROGRAMMING IN BASIC AND MACHINE LANGUAGE, by Ekkehard Floegel, published by Hofacker of Germany. This is another one of those sample books with two chapters on interfacing a Z80-PIO and relays to the ZX-81. These books can be handy once you have a good idea what to do, and just want some quick code to copy for your application.

COMPUTER COMPANION FOR THE SINCLAIR/TIMEX COMPUTERS, by Robert P. Haviland, a TAB book. This is exactly what the title implies, a support book for those needing to look up what a BASIC function means and how it is to be used. The book is in alphabetical order so you will need to know what it is you are looking for first (needs cross reference by operation desired that lists possible functions to use.) A good book to have around once you know your way about BASIC.

THE TIMEX-SINCLAIR 1983 DIRECTORY, by Eben Brown, a publication of the E. Arthur Brown Company. This is the ideal book for collectors, as it lists and displays all the products available for the ZX-81 in 1983. The keyboard chapter has eight pages devoted to all the alternative keyboards that were available. The interface section is quite impressive. Some pictures show just how far people went to interface these boards to other systems (a ZX-81 to STD Bus portable system!). At one time all the products shown were available from the authors company.

MACHINE LANGUAGE PROGRAMMING MADE SIMPLE FOR YOUR SINCLAIR & TIMEX TS-1000, by Melbourn House Publications and no author indicated. This is a book in the "Starting Forth" style, with simple and well explained steps to learning how to do assembly language programming. One good point of this book is showing two samples of each concept, as the ZX-80 and ZX-81 appear to use different registers for most operations. I think there are better books for learning assembly language (or at least newer ones), but this is very good if all your doing is ZX-81 work.

USING AND PROGRAMMING THE TIMEX SINCLAIR COMPUTER, by Ken Knecht, from Dilithium Press. This is a beginners book that actually would work as intended. The second chapter's title explains the approach, "The Computer As a Calculator." The approach is simple and straight forward. Any beginning user should do well with this book and actually understand what they are doing when finished. The book has all the right sections, including a glossary and index.

THE EXPLORER'S GUIDE TO THE ZX-81 AND TIMEX SINCLAIR 1000, by Mike Lord, from Reston Publishing Company. About 60 pages of usable insider information (Interrupts, video, block diagram) and 60 more of somewhat undocumented Basic sample programs. The insider stuff does look helpful and a few schematics are provided (16K RAM, PIO,Relay driver). This book just might fill in some facts missed elsewhere.

MASTERING MACHINE CODE ON YOUR ZX-81, by Toni Baker, a Reward Book from Reston Publishing. We have here another difficult to use book that contains probably all the facts needed to do successful assembly language programming. The explanations looks good, but unfortunately the printing is very poor (reduced 70% from full size printer output), which has been pretty common in many of the other books.

HOW TO USE THE TIMEX-SINCLAIR COMPUTER, by Jerry Willis and Deborrah Willis, from Dilithium Press. This book is a very basic approach to using the ZX-81. We have pictures showing how to hook up the machine to a TV set as an example. Not much programming, just plain old this is what you get and how it all goes together. A good getting started book for first time users.

THE SINCLAIR ZX-81, by Randle Hurley, from Dilithium Press. Another poorly printed book of reduced pages (although also available in hard bound and larger format.) A number of large Basic programs are provided, with a few pages devoted to hardware hacking. Pretty much just another ho-hum book.

BYTEING DEEPER INTO YOUR TIMEX SINCLAIR 1000, by Mark Harrison, from John Wiley & Sons publishers. A fairly decent attack at teaching simple skills needed to do Basic programing on the ZX-81. A large size book, with easy to read text (both printed and what is said), that should help any beginner understand the Basic programming of the computer.

WHAT CAN I DO WITH MY TIMEX SINCLAIR 1000, by Roger Valentine, from John Wiley Press. This large format book has very readable text, and very poor listings. The 56 programs start simple and get more complex. No real explanations, but some pointers are provided to guide you towards a better understanding of what it is your doing.

THE ZX81 TS1000 HOME COMPUTER BOOK, by David C. Foyt, and from Osborne/McGraw-Hill. A college text approach to programming the computer. Large format, with clear text, but low quality dot matrix listings. The listings are large and clean, but still somewhat a letdown when the quality of the text is so good. The book has 300 pages with over 100 devoted to appendix information. This book shows that being cheap hardware doesn't mean the book should be cheap as well. I feel this book was produced with college classes in mind and would be surprised if it had not been used in classes during 1983 or 1984.

TIMEX SINCLAIR BASIC PRIMER WITH GRAPHICS, by Mitchell Waite and Philip Chapnick, from SAMS Books/The Waite Group. The Waite Group is well know for their quality books and this is one of their early products (1984). They use charts and pictures in the "Starting Forth" style and all printing is clear and very readable in this large format book. Unlike Davids Foyt's book listed above, the listing are regular print with bold type where needed. Strickly a Basic primer, I don't think you would be disappointed starting with this book.

Well there you have the books. The last two were the best and ended up last only because they were larger sized and thus made the tall stack of books more stable. The quality of printing and listing for the most part was very poor. The text discussions varied considerably from almost none to very good even in poorly printed books.

I doubt that any of these books would be in print, so swap meets, garage sales, old book stores is your best bet of finding them. There is still considerable to be learned from them today, as they have both historical significance and good starting from scratch approaches. The main decisions for getting these books, would be whether or not you are doing ZX-81 work. Emulators abound and thus these books would help in using them (more on emulators next). The hardware might make good embedded systems especially when you might find them for $5 or 3 for $10 at a swap meet.

I am considering some sort of circuit board project based on this or the Jupiter Ace design for colleges and our readers to learn assembly programming and hardware interfacing techniques with. The advantage of using something based on the ZX-81 is the old books and emulators. These items afford the beginner many options in learning, that say Brad Rodriguez's 6809 board will not have (everything will be brand new with Brads product).

A quick explanation is needed on the "Starting Forth" reference. A book by Leo Brodie, called "Starting Forth" from Forth Inc., in 1981, is considered the best beginners book on Forth. The book uses a light hearted visual and humorous text approach to explaining the concepts of the language. This book is still being printed (second edition) and was a precursor to many of the newer explanation books. The Waite's book approach is almost identical to Mr. Brodie's and in fact some drawings look very close to being copies. I made several references to this style of approach, because many writers consider it appropriate for beginners of all levels. The reference is a form of my stamp of approval (for what it is worth).

Emulators

One reason to have a few of these books around might be to help you use a ZX-81 emulator. Since these machines do Z80 operations, you might consider developing or testing some Z80 code ideas with an emulator. This means one can be sure the design or concept is practical without ever having to built a prototype unit. Better yet is the ability to find design problems early enough in your work to prevent building something that only has to be re-done because of some over looked flaws or misconceptions.

Since one of our writers asked about the ZX-81 emulators, I loaded them from the Walnut Creek SIMTEL MSDOS CDROM and tried them out. Not being a ZX-81 super user, my first attempts were a bit baffling. As usual, I solved most of that by reading the included documentation. Each of the emulators provided considerable information, but not all of it was in English. As it turned out, what I thought was the best emulator, most of the manual is in Dutch.

The SPEC145.ZIP file contains the ZX Spectrum emulator by G.A. Lunter. These are Shareware programs and if you decide to use them, a nominal fee of $25 is requested. Actually 4 emulators are on the disk for the ZX-81. Two are the JPP version (one for VGA displays), while the last is probably a very early version used to make the others.

All these emulators are very similar and you should try them all. The JPP.ZIP has several programs that can be used with all the emulators. I found the SPEC145 to be the most complete and seemed to work better than the others. One option you need to look at is the DIAGRAMS.Z80 in the SPEC145 emulator. This program which is loaded using the F10 menu displays two schematics for using tape drives with the PC's parallel port. The interface software is available with the emulator you get when you pay the money.

Here is the list of emulators available on the CDROM. I listed all of them to give you a look at what else is available.

```
Directory MSDOS/EMULATRS/
Filename   Type Length  Date   Description
==============================================
22NCE132.ZIP   B   47988  910610  Z80 CP/M emulator for MS-DOS systems. SYDEX
68EM10.ZIP     B   56402  910615  6800 emulator for DOS, includes a realtime O/S
AME86.ZIP      B   22762  880803  CP/M-86 emulator for MS-DOS
APL2EM.ZIP     B  194206  900808  Apple ][ emulator for 286+, w/ASM source code
C64.ZIP        B   45293  921203  Commodore 64 emulator, req Hercules video card
JPP.ZIP        B  257718  921216  Emulates ZX Spectrum 48K RAM Z80-based micro
JPP_B4.ZIP     B  116085  920502  Sinclair ZX Spectrum 48K emulator, req 386+VGA
MCX11V15.ZIP   B   68826  920312  MC68HC11 MicroController multitask eXecutive
MYZ80103.ZIP   B   70573  921106  Simeon Cran's Z80 CP/M and Z-System emulator
SIM68102.ZIP   B   68653  920312  Motorola 68HC11 MicroController simulater
SPEC145.ZIP    B  149446  921111  Emulates a Sinclair ZX Spectrum on any PC
V2080J88.ZIP   B   19631  880118  Run CP/M-80 programs on system with V20 CPU
V20BOOT.ZIP    B   77965  900522  Turbo Pascal source code for V20 CP/M emulator
XTNDR093.ZIP   B  133908  930203  Timex/Sinclair ZX81 (TS1000) emulator, v0.93
Z80MU52B.ZIP   B   87532  900108  CP/M (Z80 processor) emulator for MS-DOS
ZRUN321.ZIP    B   26799  881216  Z80 CP/M emulator for MS-DOS
ZSIM20.ZIP     B  313412  921204  Z80 emulator + CP/M-80 BIOS to run CP/M
```

As you can see there are several Z80 emulators you might want to look at. The CP/M Z80 emulation comes without the overhead of the ZX-81 BASIC and comand interface. One advantage the JPP emulator provides, is the ability to use several ROMs. I have not tried the Forth Jupiter ACE ROMS yet, but will by next issue. This ability to test ROM code gives these emulators some advantages that other emulators do not.

I have used a few of the Z80 emulators and none of them allow for ROM emulation. The CP/M emulators are mainly to use CP/M software and not test hardware or ROM variations. The ZX-81 units seem to take a more hardware oriented view to their emulation. There are some emulators running around for 6800's that provide plenty of options and register or I/O displays. These are very handy when doing embedded controller design. This point is why I feel a more detailed look at the ZX-81 emulators is needed. On the surface they look very helpful, but under heavy use this feature may not prove as useful.

Some DOC's

Here is a little of the SPEC145 documentation (how's your Dutch..).

Sinclair ZX Spectrum Emulator V1.45 - 14/8/92 - door G.A. Lunter

---------------

As much as possible of the Spectrum Emulator package has been translated into English. Unfortunately it was not possible to translate the full manual. A summary of the most important things can be found in the file SUMMARY.DOC. This file LEESMIJ.DOC summarizes the Dutch manual for Dutch readers.

G.A. Lunter

--------------

In dit bestand worden de volgende vragen beantwoord:

1. Waaruit bestaat het pakket?
2. Hoe kan ik met dit programma werken zonder de documentatie te lezen?
3. Hoe zit het met verspreiding, registratie en dergelijke?

1. DE SINCLAIR ZX SPECTRUM EMULATOR

Het Shareware-pakket bestaat uit de volgende negen bestanden:

Z80.EXE - De Spectrum Emulator
ROMS.BIN - Verschillende ROM's
GETRS.COM - Utility voor overzetten van Spectrum naar PC
SAVESPEC.BAS - Idem, zie documentatie
DIAGRAMS.Z80 - Schema's voor tapeinterfaces, en calibratieprog.
Z80.DOC - Documentatie
NEW.DOC - Wat is er nieuw in deze versie? (engels)
SUMMARY.DOC - Een samenvatting van Z80.DOC voor engelstaligen.
LEESMIJ.DOC - Dit bestand

Als je je registreert, krijg je bovendien de volgende bestanden:

CONVERT.EXE - Converteerprogramma Spectrum <--> PC (plaatjes, BASIC, tekstbestanden, ...) Zie documentatie

Z80DUMP.COM - Programma om de header van .Z80 bestanden te bekijken,

CONVZ80.COM - Converteerprogramma van .Z80 files naar .SNA en .SP files van resp. VGASPEC en JPP en vice versa,

DISCIPLE.EXE - Utility voor het lezen van bestanden en snapshots van DISCiPLE diskettes,

de geregistreerde versie van de Spectrum Emulator, Z80.EXE, waarmee je Spectrum programma's direct van cassette kan laden, en weer saven, en verder de volledige sourcecode van de Spectrum Emulator en bijbehorende utilities.

2. KORTE UITLEG VAN DE SPECTRUM EMULATOR

De Spectrum Emulator 'Z80' is een programma dat een Sinclair ZX Spectrum 48K model 2 of 3, met Interface I, emuleert, met complete emulatie van de Z80, het scherm, het toetsenbord, de interrupts en het geluid, volledige cassette-ondersteuning (normaal en snellaad), en heeft vele extra mogelijkheden zoals het omleiden van de RS232 in- en uitvoer naar PC-bestanden, de printer of de COM poort, vele Spectrum joystickinterfaces die bestuurd kunnen worden door het toetsenbord, en analoge en digitale joysticks, conversieprogramma's die spectrum schermpjes kunnen omzetten in GIF-files, Spectrum BASIC-programma's kunnen omzetten in tekst en omgekeerd, die de 'snapshot' bestanden van VGASPEC en JPP (ook Spectrum emulatoren) kunnen omzetten in .Z80 bestanden (het fileformaat van deze Emulator) en omgekeerd, een leesprogramma voor DISCiPLE diskettes, een uitgebreide handleiding, en meer!

De Spectrum Emulator wordt opgestart door bij de DOS-prompt "Z80" in te tikken. Er wordt nu een Spectrum met Interface 1 opgestart.

Het programma werkt op Hercules, CGA, Plantronics, EGA en VGA kaarten. In de meeste gevallen zoekt het programma zelf uit welke kaart aanwezig is. Met een 'switch' is deze mogelijkheid uit te schakelen en kun je zelf vertellen welke videokaart gebruikt moet worden. Tik "Z80 -?" in om te zien hoe dit moet. Als je Hercules gebruikt, is het aan te raden om optie -x te proberen; hiermee krijg je op sommige monitors een mooie, grote weergave. Deze videomode noem ik Extended Hercules.

Enkele toetsen hebben een speciale functie binnen de Spectrum Emulator:

CTRL	= Symbol Shift
ALT	= Symbol Shift
F1	= Hulpscherm
F2	= Save programma (.Z80 file)
F3	= Laad programma (.Z80 file)
F4	= Verander instellingen
F5	= NMI
ALT-F6	= Reset
F7	= Real Mode
F10	= Hoofdmenu
Pijltjestoetsen	= Joystickbesturing
TAB 5 0 .	= Vuren (5 0 . op cijferblokje)
CTRL-BREAK	= Stoppen

Je kunt ook stoppen vanuit het hoofdmenu (F10). De toets F5 genereert een NMI (Non-Maskable Interrupt), en is alleen van belang als bij het opstarten optie -s is gebruikt (Zie documentatie). Met ALT-F6 wordt het Spectrum-geheugen schoon geveegd en kan bijvoorbeeld een Basic-programma ingetikt worden.

Sinclair ZX Spectrum Emulator V1.45 - 14/8/92 - by G.A. Lunter

For those who do not know enough Dutch to read the manual, Z80.DOC, a summary follows in English:

1. Introduction & Registration

The Spectrum Emulator 'Z80' is a program that emulates a Sinclair ZX Spectrum 48K model 2 or 3, with Interface I, complete Z80, screen, keyboard, interrupt and sound emulation, full tape support (normal and speedload), and has many additional features such as Interface I RS232 in/output to PC devices (file, COM, LPT), many Spectrum joystick interfaces controllable by PC cursor keys, analogue and digital PC joysticks, conversion programs that make GIF files out of Spectrum screens, turn Spectrum BASIC programs into PC textfiles and vice versa, convert snapshot files of VGASPEC and JPP (also Spectrum Emulators) into .Z80 files (this Spectrum Emulator's snapshot format) and vice versa, a DISCiPLE disk read program, an extensive manual, and more!

The Spectrum Emulator is a Shareware-program. You are allowed to try it for a month, and if you like it you should register for it. When registered, you will receive a registered version of the Spectrum Emulator, four utilities and the sourcefiles of the Emulator. The shareware-version of the Emulator is fully functional, except for the Tape support and the possibility of slowing down the Emulator, which are available only to registered users. Registered users will also be informed of future updates of the Emulator.

The shareware package of the Spectrum Emulator consists of the following files:

Z80.EXE - The Emulator
ROMS.BIN - The ROMs of the Spectrum and SamRam
GETRS.COM - Utility to send programs from Spectrum to PC
SAVESPEC.BAS - Idem, see paragraph 6.
DIAGRAMS.Z80 - Diagrams for tape interfaces, and calibration
Z80.DOC - Documentation file -- in Dutch
LEESMIJ.DOC - Summary of documentation -- in Dutch
NIEUW.DOC - What's new
SUMMARY.DOC - This file

These are the utilities you'll receive upon registering:

CONVERT - a general conversion program for converting many Spectrum file formats into PC formats and vice versa. It can produce GIF files of Spectrum screens, make text-listings of programs and more.

CONVZ80 - Translates snapshot files from VGASPEC's .SNA and JPP's .SP formats to this Emulator's .Z80 format and vice versa.

Z80DUMP - Shows the contents of the header of a .Z80 file

DISCIPLE - Reads DISCiPLE diskettes. It translates normal files into files that can be read using the Interface I's LOAD *"b" command, and translates snapshot files into .Z80 files.

The registration fee is 35 Dutch guilders, 10 English pounds, 20 US$, oder 35 Deutsche Marken. For Dutch users, the registration fee is Hfl 25,--. Please send the money to:

G.A. Lunter,
Aweg 11a,
9718 CT Groningen,
The Netherlands

If you can, please use Eurocheques or send it cash. If you use Eurocheques please do NOT fill in the place (or fill in 'Groningen'), because otherwise I have to pay a large commission. Unfortunately I cannot accept creditcards. Also, please don't forget to give me your address (including your country.) It is also possible to reach me by Internet via Johan Muizelaar: his e-mail address is mouse@frw.rug.nl

Final Comments

I did also look at the Walnut Creeks CDROM of Source Code under MSDOS/EMULATORS and found the Turbo Pascal source for 20Boot, a CP/M emulator. (Also an Apple][emulator 8088 ASM code.) Under MSDOS/ASMUTIL is a Z80-22 emulator. Under the USENET section is a TRS80 emulator, Apple emulator, and several Z80 emulators. So once again if you need to do something special, the Z80 has plenty of software and hardware emulators and older systems, all available for nothing to practically nothing.

If you don't have a CDROM yet, many of the neworks have the emulators or can get them. I think JW Weaver's BBS has the Walnut Creek MSDOS CDROM on line all the time now. So check your local BBS and ask around.

I expect to hear more from you and will be looking the Jupiter Ace over next. The Jupiter was a Forth based advanced version of the ZX-81 and might just be ideal system for new projects.

32-Bit Systems

All Readers

Linux & Linking

Real Computing

By Rick Rodman

Windows NT

Windows NT was very easy to install. It automatically recognized the boards in my system, including the Western Digital Ethernet board, and includes a version of TCP/IP. It runs quite well on a 386/33 with 16 megabytes of RAM, taking about 90 megabytes of hard disk itself. Since I consider it a competitor to OS/2, I don't think these requirements are unreasonable.

It's nice that the TCP/IP comes with NT, because not much else does. It looks and feels like regular old Windows, and most Windows programs will run, unless they have any hardware drivers. And these are the sources of the main problems with NT. I don't have much use for an operating system with no device driver support. IBM has included everything you need to write device drivers, and more, with the OS/2 Toolkit. Microsoft, however, has always separated device drivers into separate ''Device Driver Kits'' for Windows, DOS, and now NT. Since device drivers are a part of life, I disagree with this philosophy. Surprisingly simple programs can come up needing device drivers, and the presence or absence of particular device drivers may be a factor that will make or break NT in the marketplace.

Another problem is that NT has only a single ''virtual DOS machine'' (VDM), which emulates a 286. This makes it not very suitable for software development, where we may want several DOS windows assembling, compiling, etc. There is a known problem with debugging DLLs under Windows: a DLL, if the program which loaded it crashes or fails to unload it, cannot be cleared without exiting Windows. This problem exists under NT also, but is even more annoying there. Logging out and back in seems to restart the VDM, but I'm not sure. I'm not convinced, either, that an errant Windows program cannot crash the entire operating system.

Under OS/2, by contrast, there are multiple VDMs, which can be closed and reopened at will. A totally trashed Windows session can simply be killed and restarted without affecting anything else. Windows programs may run a little faster under NT than under OS/2, but NT's single VDM will limit how many you can run. NT-native apps may ease these problems, but even then, it doesn't seem like you'll get much more capability than you had under regular old Windows.

NT has a long ways to go. But remember, Microsoft has gone a long ways in the past.

E-mail fun

There's a lot more to e-mail than meets the eye. Gary Welles wrote again and sent a lot of helpful information on X.400 and MCI Mail. If you would like these files, e-mail to me and I will mail them back to you.

Gary's MCI Mail address is Gary_Welles/117-8863@MCIMail.com, but his X.400 address, he says, would be ''G=gary; S=welles; DDA=ID=1178863; A=mci; C=us''. The DDA part is a ''domain-defined attribute''; the A part is the Administrative Domain.

Gary encouraged me to try the FTP mail server (ftpmail@decwrl.dec.com) again. I came across some mention of programs to read Photo-CD files and convert them into PPM raster format. My main interest was in understanding the Photo-CD format. At any rate, by using a search program called ''archie'', a correspondent informed me that the file was located on one computer in Denmark and another in Japan. So, without being too hopeful, I sent the mail message to retrieve the file from the computer in Denmark.

Ten minutes later, I had the file. Astonishing.

Please note that my e-mail address has changed. It's now rickr@aib.com.

Tiny TCP

In our last issue I discussed the concept of a ''little network'' connecting a variety of small machines, and the small TCP/IP packages I knew of at that time. Because of the size of those packages - the ''old'' KA9Q, for example, is around 15,000 lines - I figured the effort required to successfully port them to more than one or two machines would be too high to justify it.

I've run across another small TCP/IP package - very small, in fact. It's called Tiny-TCP, and it claims to have been written in three days by Geoffrey Cooper at Imagen Corporation. It includes FTP receive file (only) logic and a driver for an old 3Com Multibus board (only). Originally it seems to have been intended to be burned in PROM. It's only 1439 lines of source in C as delivered.

This appears to be a good foundation for our little network. I've written a SLIP (Serial Line Internet Protocol) driver for the package, so that it can use the serial port rather than an Ethernet board, and this is working. The next step will be to complete the FTP application parts so that it can transfer files back and forth. So far, the source size has only grown to 2169 lines.

After this, I can port it to the PC-532, Amiga, DEC Rainbow, Xerox 820, and whatever machines I need to connect. It may be necessary to add routing capabilities to the package, and to expand the SLIP logic to support multiple ports. Hopefully this can be done without bloating the source too much. I have no desire to add ICMP, UDP, SMTP, SNMP, PPP or any of the rest of the TCP/IP alphabet soup into the mix.

If you'd like to share in the fun, send me an e-mail message and I'll e-mail the code back to you as a reply. Alternatively, call the BBS, or send a disk with a reusable mailer. (This policy applies to Bare Metal and most other software discussed in this column, too.)

Incidentally, I found this code on the Walnut Creek Source CD-ROM. (Can we just start saying "CD" and leave off the "ROM"?) There's so much stuff on that CD that it's hard to find things. They need some kind of keyword search.

More on Linux

A Linux magazine has been started in England. I don't know the name or address, however. Perhaps one of our readers in England can let me know.

I mentioned last time that work was underway to port Linux to the Amiga. I don't know the status of that port. Also, the public-domain BSD operating system, "NetBSD", is being ported to the Amiga. If you're interested, and have e-mail access (remember, if you have Compuserve, MCI Mail, AOL, or FidoNet, you can use a gateway), you can subscribe to the Amiga "netbsd-amiga" mailing list by sending a message to "netbsd-admin@cbmuucp.commodore.com". You can get binaries for it by ftp (or ftpmail) from ftp.eunet.ch, directory software/os/bsd/NetBSD, file NetBSD-Amiga.

In #64 I mentioned that I had loaded Linux from a CD-ROM called "Yggdrasil" (which is the name of a mythical ash-tree at the center of the earth, or some such nonsense). There are other ways to obtain it. Some have price tags attached to them, but remember, nothing is free - not network time or long-distance telephone, surely.

For those without a CD-ROM drive, a diskette version has been put together which can be loaded from 3.5", 1.44 megabyte floppies. This package is called "Slackware", because it's supposed to be very easy to install. Because Linux is so huge with all of the add-ons you can get, it has been broken into disk sets A, E, F, G, Q, and H. There is really an astonishing amount of code here; the complete set amounts to 31 diskettes.

Now if you have the time to transfer over forty megabytes of data through your modem, you can ftp the files from ftp.cdrom.com, directory /pub/linux/slackware. Also, they are available with Zoo compression in /pub/linux/slackware/zooed_disks. At 9600 baud, you should be able to transfer this much data in about a day.

On the other hand, if your time has any value, you might want to order diskettes from a gentleman named Stephen Balbach. He asks for $41 for 5" disks or $56 for 3-1/2" disks. Also, Matt Welsh has written a book called "Linux Installation and Getting Started", which he is apparently self-publishing at a cost of $20. This is also available from Steven Balbach.

Next time

I haven't had a chance to look at Sprite yet, so I hope to have that, and networking stuff on Linux, Tiny-TCP, and more E-mail stuff. There are more stirrings on the PC-532 front, too. So much to do. But the Real Computing motto is: It can be done. And if it's already being done, it can be done faster!

Where to call or write

Real Computing BBS or Fax: +1 703 330 9049
E-mail: rickr@aib.com
Mail: 8329 Ivy Glen Court, Manassas VA 22110

Slackware Linux:
E-mail: linux-slack@clark.net, or stephen@clark.netMail: Stephen Balbach, 5437 Enberend Terrace, Columbia MD 21045Phone:+1 410 740-1157

Special Feature

Intermediate Users

Reset Circuits

Little Circuits

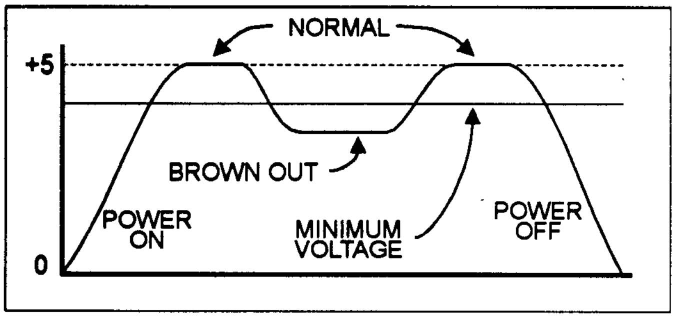
by Dave Baldwin

This is about common hardware problems and solutions. Most of the solutions come from the research and experiments I've done after I've had the problem. It's interesting that many of the problem circuits have analog aspects to them. Microprocessor examples will based on the Z80 since that is what I'm most familiar with.

Since this is the beginning, it seems that reset circuits should come first.

BASIC CONDITIONS

Reset circuits in digital and microprocessor systems are important at four different times: power on, power off, brown-out, and manual reset. Well designed reset circuits make sure the system starts, stops, and restarts properly. Reset circuits can range from a single RC network for a microcontroller to several IC's for larger systems.



At power on, the reset circuit sets up the initial operating conditions and delays system operation until the clock oscillator and other sub-circuits are operating properly. When power is turned on, most circuits come up randomly without the assistance of a proper reset signal. Crystal oscillators can take several hundred milliseconds for the output to come up to full output voltage after power on.

At power off, the reset circuit should halt the system before the power supplies drop low enough to cause improper operation. Strange things can happen as the power supplies drop unless the reset circuit prevents it.

Brown-out (when the power drops momentarily) is a combination of power off and then power back on again. Simple systems just do power off and power on resets as the supplies change. Fancier systems use early detection of the power supply drop to do a 'halt and resume' with a combination of hardware and software.

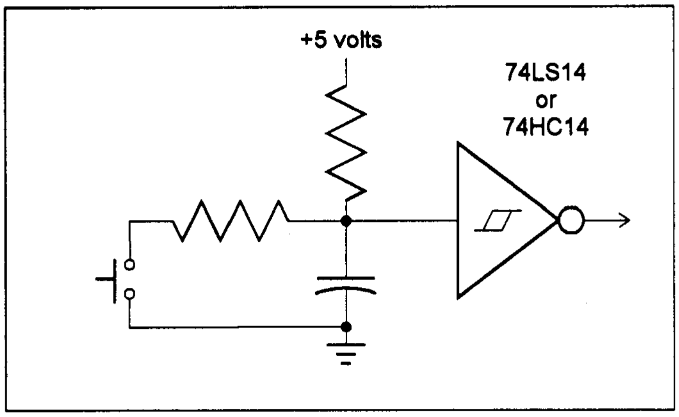
Manual reset during normal operation is used to restart a system. Desktop systems need to be manually reset when a program hangs up or when the system needs to be restarted with a new configuration. Small microcontrollers are frequently designed so that the manual reset input acts as the 'start' button for whatever function they perform.

POWER ON CIRCUITS AND CAPACITORS

Circuits that depend on charging a cap to generate a reset pulse at power up also depend on how fast the power supply rises. If your power supply voltage comes up real slow, then the cap can charge just as fast as the voltage rises and you never get a reset pulse. My Big Board 1 did this until I increased the value of the capacitors. This can sneak up on you if you test your system on a bench supply that comes up nice and fast and then put your unit in a box with a slow wimpy power supply that is too slow to activate your reset circuit. (I've done it. Causes a lot of head scratching. After all, ''It used to work!'' and will if manually reset after power has been on for a few seconds.)

Reset circuits usually use a Schmitt trigger at the input to get a clean signal at the output. A Schmitt trigger has more gain (analog term) than a regular gate, different switching thresholds for positive and negative going signals, and positive feedback to make sure the output switches completely when the input switching threshold are crossed. The 8048 and 8051 microcontrollers have this built in so they only need a capacitor or a resistor and capacitor for a power on reset.

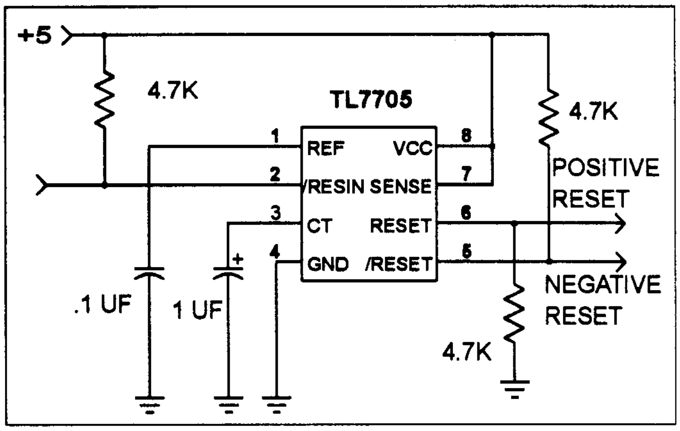
An RC network connected to the input of a 74LS or 74HC14 can serve as a simple power on and manual reset circuit. It won't do anything useful at power off though. A problem with 74LS or any other versions of TTL is that they require almost 4 volts before they will operate properly. This means that other parts of your system may be active before a proper reset signal occurs. 74HC parts are supposed to operate down to 2 volts.



If you are going to use a switch for manual reset, you should always use a resistor in series with the switch and the capacitor to limit the current out of the cap. Shorting a capacitor directly to ground can cause excessive ground currents and system failure or erratic operation. (A 10uf cap charged to 5 volts and then discharged through a switch to ground in 1 usec means a peak current of 50 amps!)

POWER ON AND OFF

Circuits that do both power on and off resets require a way to check the power supply voltage against a reference. The circuit should hold the reset line low until the power supply voltage is high enough for normal operation. It should also bring the reset line low any time the power supply drops below the required voltage.



The TL7705 from Texas Instruments is an IC that includes all of this in an 8-pin Dip package and provides a manual reset input along with hi and lo reset outputs. It requires a cap to set the reset time constant and one for a filter along with a couple of resistors for pull-up/down. The TL7705 is designed for 5 volt supplies. The TL7702 uses another resistor to set the input threshold for any positive supply voltage. The reset outputs on the 'B' versions are active as soon as the power supply reaches 1 volt.

Circuits that use an RC network to set the reset pulse width assume that the clock oscillator output will be at full voltage before the end of the reset pulse. You can add a counter to your reset circuit to guarantee a certain number of valid clocks before the reset signal is released. The Big Board II uses a 74LS393 to require 8 clock cycles before the reset pulse is released. Although it does have the problem of a TTL circuit, it does require the cpu clock to be operating before the reset is released.

The Big Board 1 and II have a special circuit for the manual reset that delays the reset signal until the beginning of an instruction fetch as signalled by the M1 signal from the Z80 cpu. This prevents memory writes and I/O cycles from being disturbed by the manual reset. The extra circuits are needed because the Z80 acts on a reset signal immediately.

BROWNOUT AND RESTART

A different reset configuration is required if you want to temporarily halt and then resume operation when a brownout occurs. This requires a power failure circuit that will signal the cpu (usually thru a non-maskable interrupt) before the reset circuit stops the cpu completely. When the power failure interrupt occurs, the cpu must execute a software routine that saves the current conditions so they can be restored when power is good again. This can be in a battery-backed real-time clock or ram. The MAX693 from Maxim is one chip that provides all the functions above plus it has a battery switch that provides back-up power to a real-time clock or ram and a 'chip enable' control circuit that prevents power down glitches from corrupting the data in the chip.

RESET SYNCHRONIZATION

Many CPU chips require the reset signal to be synchronized with the CPU clock signal. The clock driver chips for the X86 processors include a flip-flop for this function. You must provide this function if you design your own reset circuits for CPU's that require this.

NEXT TIME

Next time I'll cover battery-backup circuits for real-time clocks and rams, another function that has caused many problems.

Regular Feature

Classic Support

Group Reviews

SUPPORT GROUPS FOR THE CLASSICS

By JW Weaver

Activity on my BBS is increasing by TCJ readers, Please leave messages on subject(s) you are interested in finding on the BBS, or hints on improvements you feel are needed. Numerous changes have taken place just in the last few weeks, so read the *log on* message for a list of changes and new options.

Received a message from Tilmann Reh, giving information on some projects, and a suggestion for subject classification of projects. I am going to implement his ideas in this issue.

PROJECT CATEGORIES:

Hardware - In progress
Hardware - Completed
Hardware - Needed
Software - In progress
Software - Completed
Software - Needed

For the "In progress", if help is needed or questions need answering, send to this column and I will include your information for other to see and hopefully to help get you some answers.

For the "Completed", any history on why you started this project, problems encountered during construction, is the finished project performing as expected, will you be making improvements, if project is available to other readers, and how may the other readers obtain information on, or get the completed project.

For the "Needed", I will give names and contact procedure for volunteers. I will keep any on-going information on my BBS for up-to-date inquires.

Memory Problems

During the testing of a memory expander, for a Commodore 64, the first 768 Kb of memory tested valid, but the remaining 768 Kb returned scrambled data. The individual memory chips tested good, so this indicated something was NOT quite right with the support circuitry. A visual check did not indicate any bad connections or mis-wiring. So how do we test this project without elaborate test equipment? Well, what about building an add-on circuit for a computer to do the checking!.

Looking around at the many computers I have available, which one will be the easiest to design for and program... there sits my lowly Kaypro 4, simple, 2 floppy drives, 64 Kb of memory, serial port, parallel port, and an internal expansion connector with all the processor signals.

First off, this Kaypro 4 uses the 81-184 motherboard, but almost any Kaypro with an expansion connector (J9) can be used.

Constraints imposed by the Kaypro, are these:

I/O addressing scheme - ports 01 thru 24 and 80 thru 85 are used (or are defined for use) by system hardware. This leaves addresses 30 thru 7F as clear blocks for port definition.

Expansion connector - is internal so must be brought outside to be useful. Two of the signals required are inverted logic and will require inverters to correct logic levels.

Nothing too complex yet, so off we go.

Choosing the addresses 30 thru 3F will allow 16 ports for usage, and allow adequate I/O lines, the lines needed are 8 output and 6 inputs, and can be handled by any device with 2 8bit ports.

The functions that must be solved are a) address selection b) data I/O c) logic level correction. The IC (IC1) to fill the function "a" is the 74138, a 1 of 8 decoder with 3 enable lines. The IC (IC2) to fill the function "b" is the Z80-PIO chip, this gives two separate 8-bit I/O ports and only require 4 consecutive addresses. The IC (IC3) to fill the function "c" is the 7404, six inverters in one package.

The addresses given for the I/O port are processor address bits 7 thru 0, with the ports desired 30 - 33, the bits 7 thru 4 pattern looks like 0011 which give the 3x group of I/O addresses, to tie this to IC1 correctly, address bit 7 is tied to IC1 Enable 3, address bit 6 is tied to IC1 Enable 2, these two enables are inverted internal to the IC1 before being fed into a AND gate with Enable 1, address bit 5 is tied to IC1 Enable 1, address bit 4 tied to IC1 Address in 2, address bit 3 tied to IC Address in 1, address bit 2 tied to IC1 Address in 0, yield valid I/O addresses 2x thru 3x. This chip selects 1 of 8 outputs, each output selects an I/O address range of Output 0 equates to 20 thru 23,

Output 1 =	24 thru 27,
Output 2 =	28 thru 2B,
Output 3 =	2C thru 2F,
Output 4 =	30 thru 33,
Output 5 =	34 thru 37,
Output 6 =	38 thru 3B,
Output 7 =	3C thru 3F,

Output 4 is tied to IC2 CE, to get the I/O addresses 30 thru 33.

With address bit 1 tied to IC2 Command/Data and address bit 0 tied to IC2 A/B device port. This completes the address lines. The logic level inversion function is performed by IC3, taking Kaypro line RD is tied to IC3 Input 1 with Output 1 tied to IC2 RD, line IORQ tied to IC3 Input 2 with Output 2 tied to IC2 IORQ.

Wire connect list is as follows:

J9	IC3	IC2	IC1	Function
2			19	D0
4			20	D1
6			1	D2
8			40	D3
10			39	D4
12			38	D5
14			3	D6
16			2	D7
18		4		E1
20		5		E2
22		6		E3
24		3		A2
26		2		A1
28		1		A0
30		5		C/D
32		6		A/B
34	3			IORQ
36			37	M1
40	1			RD
44			23	Interrupt
49			25	Clock
2			35	RD
4			36	IORQ
11			4	O4
14		16	26	+5
7		8	11	Ground
15				port A0
14				port A1
13				port A2
12				port A3
10				port A4
9				port A5
8				port A6
7				port A7
27				port B0
28				port B1
29				port B2
30				port B3
31				port B4
32				port B5
33				port B6
34				port B7

1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31
are Ground
33,35,37,39,41 are +5
IC1-24 tied to Ground
IC3-5,9,11,13 tied to 1000ohm to +5

Expansion is readily accommodated by the additional selects of IC2. In theory, a total of 6 Z80-PIO's could be attached. If you do expand this circuit, be sure to buffer D0 thru D7 and A0 thru A7.

To initialize the PIO(s), each port must be set to the desired functions. In my case the code outputs bytes 4F and 00 to I/O address 32 to set Port A to an output function, with bytes 0F and 00 to I/O address 33 to set Port B to an input function. The use is by outputing a byte to I/O address 30 to set conditions needed, with an input of a byte from I/O address 31.

Add a user interface and you have the beginnings of a computer controlled test system. If readers are interested about more information, i.e., code or schematic, they will be posted to my BBS under Applications.

Keep Hacking, JW Weaver.

Contact by US Mail
TCJ Support Group
Drawer 180
Volcano, California

Contact by BBS
(916) 427-9038
up to 2400 baud 8 bits N parity 1 stop

Terms Used
Kb Kilobytes
I/O Input/Output
IC Integrated Circuit
Notes: All numbers used in regards to addressing are in hexadecimal.

PROJECTS

Tilmann Reh.

General activities can be described as:

- Modular computers, especially Z180 and Z280 and based on ECB bus;
- GENERIC CP/M-3 (-Plus) software;
- embedded control with microcontrollers.

Software, finished:

- Generic MSDOS disk emulator for CP/M-3
- Generic MSDOS disk reformatter for CP/M-3
- Generic directory utility for CP/M-3 (4DOS-like)
- Generic harddisk PARKing utility for CP/M-3
- ARC V2.1 which runs with both CP/M 2.2 and 3

Software, needed:

- Generic SCOPY for CP/M-3 (already asked R.Friefeld)
- Generic GOOD (!) Backup software for CP/M-3 (I already have some ideas, but not enough time)
- Generic CHKDSK utility for CP/M-3 (checking the validity of the complete directory)
- Somebody please BUGFIX the disk utility DU90! (there are several hard bugs)
- Turbo-compatible Pascal compiler (maybe very slow!) which generates good optimized code. (I would like to do the final compiling of my programs with such a compiler!)

Contacts needed:

Searching for the author of the PMARC program package (containing PMARC, PMEXT and some utilities) The author is named Yoshihiko Mino. Perhaps anyone of the TCJ readers is able to help?

TCJ Staff Contacts

TCJ Editor:
Bill D. Kibler, PO Box 535, Lincoln, CA 95648, (916)645-1670, GEnie: B.Kibler, CompuServe: 71563,2243, E-mail: B.Kibler@Genie.geis.com.

Z-System Support:
Jay Sage,1435 Centre St. Newton Centre, MA 02159-2469, (617)965-3552, BBS: (617)965-7259(pw=DDT), MABOS on PC-Pursuit, E-mail: Sage@ll.mit.edu. Also sells Z-System software, see inside front cover.

32Bit Support:
Rick Rodman, BBS:(703)330-9049 (eves), E-mail: rickr@virtech.vti.com.

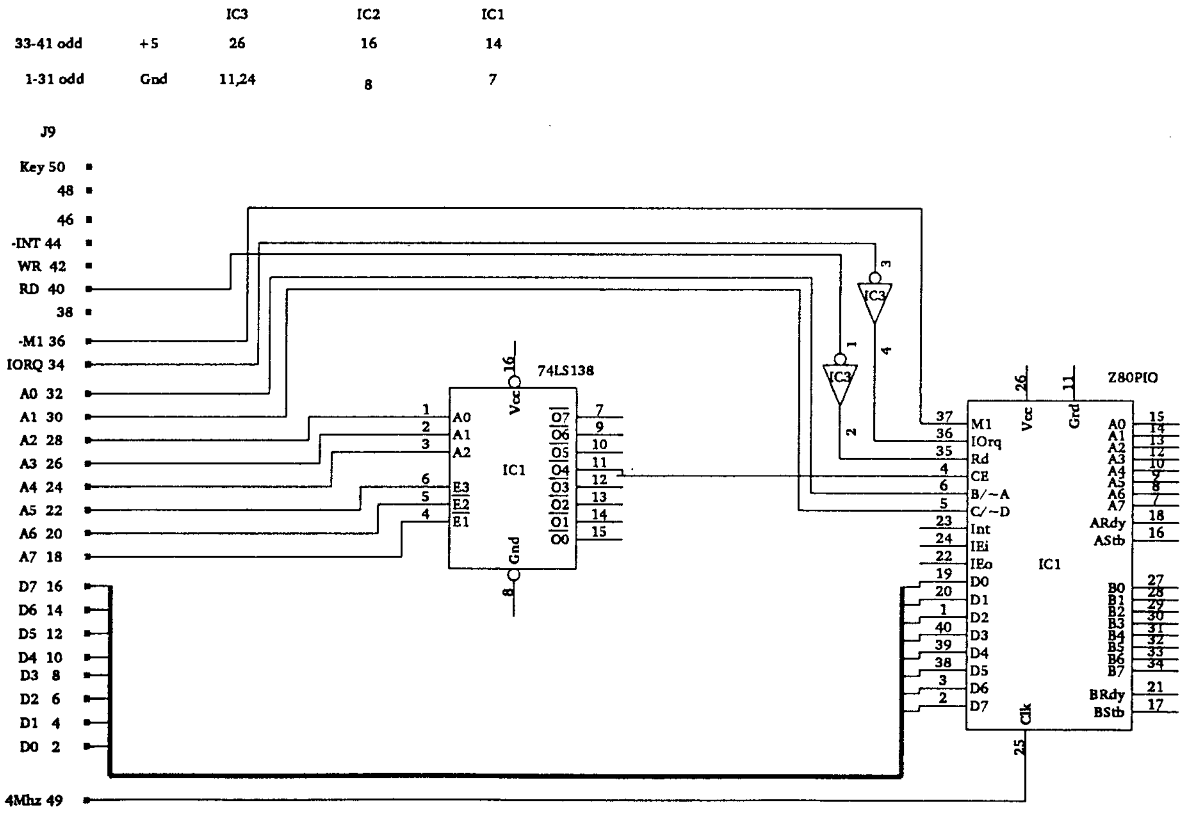
Kaypro Support:
Charles Stafford, 4000 Norris Ave., Sacramento, CA 95821, (916)483-0312 (eves). Also sells Kaypro upgrades, see ad inside back cover.

S-100 Support:
Herb Johnson, CN 5256 #105, Princeton, NJ 08543, (609)771-1503. Also sells used S-100 boards and systems, see inside back cover.

6809 Support:
Ronald Anderson, 3540 Sturbridge Ct., Ann Arbor, MI 48105.

Users Groups and Project Reports:
JW Weaver, Drawer 180, Volcano, CA 95689, BBS: (916)427-9038.

Regular Contributors:
Brad Rodriguez,Box 77, McMaster Univ., 1280 Main St. West, Hamilton, ONT, L8S 1C0, Canada, Genie: B.Rodriguez2, E-mail: b.rodriguez2@genie.geis.com.



Frank Sergeant, 809 W. San Antonio St., San Marcos, TX 78666, E-mail: fs07675@academia.swt.edu.

Tilmann Reh, Germany, E-mail: tilmann.reh@hrz.uni-siegen.d400.de. Has complete MS-DOS disk emulation program for CP/M+, contact Jay Sage.

Helmut Jungkunz, Germany, "Virtual" ZNODE #51, or CompuServe 100024,1545.

USER GROUPS

Older systems:
Connecticut CP/M Users Group, contact Stephen Griswold, PO Box 74, Canton CT 06019-0074, BBS: (203)665-1100. Sponsors East Coast Z-fests.

Sacramento Microcomputer Users Group, PO Box 161513, Sacramento, CA 95816-1513, BBS: (916)372-3646. Publishes newsletter, $15.00 membership, normal meeting is first Thursday at SMUD 6201 S st., Sacramento CA.

Coleco ADAM:
ADAM-Link User's Group, Salt Lake City, Utah, BBS: (801)484-5114. Supporting Coleco ADAM machines, with Newsletter and BBS.

Adam International Media, Adam's House, Route 2, Box 2756, 1829-1 County Rd. 130, Pearland TX 77581-9503, (713)482-5040. Contact Terry R. Fowler for information.

AUGER, Emerald Coast ADAM Users Group, PO Box 4934, Fort Walton Beach FL 32549-4934, (904)244-1516. Contact Norman J. Deere, treasurer and editor for pricing and newsletter information.

MOAUG, Metro Orlando Adam Users Group, Contact James Poulin, 1146 Manatee Dr. Rockledge FL 32955, (407)631-0958.

Metro Toronto Adam Group, Box 165, 260 Adelaide St. E., Toronto, ONT M5A 1N0, Canada, (416)424-1352.

Omaha ADAM Users Club, Contact Norman R. Castro, 809 W. 33rd Ave. Bellevue NE 68005, (402)291-4405. Suppose to be oldest ADAM group.

OS-9 Support:
San Diego OS-9 Users Group, Contact Warren Hrach (619)221-8246, BBS: (619)224-4878.

Atari Support:
ACCESS, PO Box 1354, Sacramento, CA 95812, Contact Bob Drews (916)423-1573. Meets first Thurdays at SMUD 59Th St. (ed. bldg.).

Forth Support:
Forth Interest Group, PO Box 2154, Oakland CA 94621 510-89-FORTH. International support of the Forth language. Contact for list of local chapters.

OTHER PUBLICATIONS

The Z-Letter, supporting Z-System and CP/M users. David A.J. McGlone, Lambda Software Publishing, 149 West Hillard Lane, Eugene, OR 97404-3057, (503)688-3563. Bi-Monthly user oriented newsletter (20 pages+). Also sells CP/M Boot disks, software, and new versions of Borland CP/M software.

The Analytical Engine, by the Computer History Association of California, 1001 Elm Ct. El Cerrito, CA 94530-2602. A ASCII text file distributed by Internet, issue #1 was July 1993. E-mail: kcrosby@crayola.win.net.

Z-100 LifeLine, Paul F. Herman Inc., 9317 Amazon Drive, New Port Richey FL 34655, (800)346-2152. Publication and products for Z-100 and S-100 machines.

The Staunch 8/89'er, Kirk L. Thompson editor, PO Box 548, West Branch IA 52358, (319)643-7136. $15/yr(US) publication for H-8/89s.

Sanyo PC Hackers Newsletter, Victor R. Frank editor, 12450 Skyline Blvd. Woodside, CA 94062-4541, (415)851-7031. Support for orphaned Sanyo computers and software.

the world of 68' micros, by FARNA Systems, PO Box 321, Warner Robins, GA 31099-0321. E-mail: dsrtfox@delphi.com. New magazine for support of old CoCo's and other 68xx(x) systems.

Amstrad PCW SIG, newsletter by Al Warsh, 2751 Reche Cyn Rd. #93, Colton, CA 92324. $9 for 6 bi-monthly newsletters on Amstrad CP/M machines.

Other Support Businesses

Sydex, PO Box 5700, Eugene OR 97405, (503)683-6033. Sells several CP/M programs for use with PC Clones ('22Disk' format/ copies CP/M disks using PC files system).

Elliam Associates, PO Box 2664, Atascadero CA 93423, (805)466-8440. Sells CP/M user group disks and Amstrad PCW products. See ad inside back cover.

Davidge Corp. 94 Commerce Dr. PO Box 1869, Buellton CA 93427, (805)688-9598. Z80 support of Davidge and Ampro Z80 Little Board.

Star Technology, 900 Road 170, Carbondale CO, 81623. Epson QX-10 support and repairs. New units also avialble.

Star-K Software Systems Corp. PO Box 209, Mt. Kisco, NY 10549, (914)241-0287, BBS: (914)241-3307. 6809/68000 operating system and software. Some educational products, call for catalog.

Peripheral Technology, 1480 Terrell Mill Rd. #870, Marietta, GA 30067, (404)973-2156. 6809/68000 single board system. 68K ISA bus compatible system. See inside front cover.

Hazelwood Computers, RR#1, Box 36, Hwy 94@Bluffton, Rhineland, MO 65069, (314)236-4372. Some SS-50 6809 boards and new 68000 systems.

AAA Chicago Computers, Jerry Koppel, (708)202-0150. SS-50 6809 boards and systems. Very limited quanity, call for information.

MicroSolutions Computer Products, 132 W. Lincoln Hwy, DeKalb, IL 60115, (815)756-3411. Make disk copying program for CP/M systems, that runs on CP/M sytems, UNIFROM Format-translation. Also PC/Z80 CompatiCard and UniDos products.

Special Feature

All Users

Selecting a Language

LEVELS OF FORTH

by Glen B. Haydon

A Language for Beginners

At the last Forth day meeting in the San Francisco Bay area, I had a discussions with Glen Haydon of Mountain View Press. Most Forth users know Glen for his excellent books and documentation on Forth.

Glen was also involved in MVP Forth (Mountain View Press Forth) and pointed out a number of things about this version of Forth. The point that struck me the most, was NO bugs. That is correct, no bugs are known to exist in MVP Forth. It has been out for many years (I think, over ten) and several books cover it's usage.

As Forth's go MVP is not the most complex and definitely not like FPC which has everything including the kitchen sink in it. Our discussion centered on getting beginners up to speed, and as such the simpler word set and design of MVP is felt by Glen and myself to be a better match to learn on.

At this point in our conversation, Glen produced the short paper he did on the different levels of Forth users and the type of Forth that meets their needs. He gave me his permission to reprint it for our readers. I think it will help many readers understand the concepts of choosing a language (of any type) and help you start focusing on doing your programming with the task in mind and not the language. BDK.

Abstract

Match the level of Forth with the user and the application.

Introduction

Ontogeny recapitulate phylogeny. I guess it was John Dewey who emphasized that you start a student where he is. The needs of novices perhaps evolve in somewhat the same way as history. If you are already at the end, you have no place to go.

A problem people in the Forth community face is the variety of applications which lend themselves to the language and the background of the Forth users. One way to address the problems is to identify progressive levels of Forth and match them with the users and the applications. These levels recapitulate their history.

Level 0.

On several occasions C. H. Moore has listed a minimal number of functions which he considers the essence of Forth. I noted them at one presentation as follows:

```
@CODE = + - * /MOD MIN MAX AND OR XOR
NEGATE ABS NOT */ DUP DROP SWAP OVER
DECIMAL HEX OCTAL . n .R CR EMIT KEY : ;
CREATE , ALLOT IN ELSE THEN FOR NEXT I
```

There are 45 functions in this list. You can see about which period this was. The list does not include BEGIN, UNTIL, and several other functions that probably belong in Level 0. A figure of about 63 is appropriate.

Three functions are used for output to the display, but no functions for any kind of storage. Screens are not included much less any means of writing on such screen.

Something more than Level 0 is necessary to provide anyone but Chuck enough to work with.

Level 1.

The fig-Forth *Installation Manual* provides a complete set of Forth functions. The implementation with the *Installation Manual* is for the 6502. since then, over a dozen implementations of fig-Forth for other processors were written and are still available in hard copy. About forty hours of careful typing will allow one to enter a listing.

In fig-Forth, we lacked an editor. It was a chicken and egg problem. Though one had screens for compiling code there was no way to write to a screen until an editor was included. A simple single function editor could be typed in interactively that allowed one to place code on a specific line of an already listed screen.

The fig-Forth implementation took over the disk precluding its use for other programs. The error messages were stored on a disk.

The fig-Forth model served as a wonderful learning tool. There was little in the way of documentation beyond the *Installation Manual* and the source listings. A few computer teaching programs did appear. The older publications such as the *Kitt Peak Primer* and *Using Forth* do not mesh well with fig-Forth.

Excellent applications have been done in fig-Forth.

Soon there came the 79 Standard which made several changes in fig-Forth. It was then proposed that the Forth Interest Group stop supplying the fig-Forth source listings.

Level 2.

The 79 Standard made about 40 changes to the functions in fig-Forth. These made some things much easier. About the same time Leo Brodie came out with the first edition of *Starting Forth*. This is excellent book for a beginner. He developed the first part of the book in a very creative manner. Then he had to get on with the job.

At that time, Mountain View Press came out with a public domain Forth implementation incorporating the 79 Standard changes. If the Forth Interest Group was going to withdraw their listings from the market, the Forth community deserved something to take its place.

MVP Forth puts the error messages in line and adds supplemental functions. All but one of the functions matched those covered in *Starting Forth*. The single problem is with 'tic'.

A Forth assembler for each implementation is included. The Forth editor described in Chapter 3 of *Starting Forth*, written by Sam Daniels, is included. Note: Starting Forth does not point out that you have to invoke the EDITOR vocabulary for it to work!

The complete source listing of MVP Forth is available for three processors. The appropriate source is included on each distribution disk. The Glossary, *All ABOUT FORTH* 2nd Edition, includes the functional definitions, source code for the 8080, examples and comments for each of about 240 functions. The 3rd Edition of *ALL ABOUT FORTH* has been expanded., The book is a complete Forth reference manual with a concordance of seven functional definitions and four implementations for 500 functions now in common usage. The appendices include the current Standard documents.

The distribution Version 03 of MVP Forth has remained unchanged since April 1983. Unfortunately several copies which have appeared on bulletin boards that, while being designated MVP Forth, have been modified. The original Version 03 is now available on the GENIE Forth Round Table and the Mountain View Press bulletin board.

The MVP Forth implementation takes over a disk as does fig-Forth. But the file on GENIE includes several added utilities which allow one to access screens in files and compile text files written with any word processor.

Most of the Forth text books and tutorials begin with a discussion of stacks. After all it is the stacks which makes Forth what it is. But for most beginners stacks are a complication and a distraction. In the *FORTH GUIDE*, a different introduction is used.

The hardware designers and engineers have other interests besides learning yet another language. A small language which allows them to get on with their work is sufficient. Levels 1 and 2 are closely bound to the hardware and serves such users well.

Level 3.

Several developments led to a new implementation. The 83 Standard was finally adopted. Few people had computers dedicated to Forth. They needed their computer for other things. There was also a growing library of Forth functions which could be incorporated.

F83, by Henry Laxon and Mike Perry, took the next step. F83 includes nearly 1000 functions. A substantial library of Forth functions was added to the 83 Standard implementation. Ting's book, *Inside F83*, provides documentation.

The most popular implementation of F83 is for Intel based systems. It makes use of screens but now in files. A dedicated disk is no longer necessary and programs could be developed and run from a hard disk. Many of the functions are optimized in assemble code which improves the speed. With the library of functions, we were on the to an implementation of Forth for programmers.

But F83 with its increased vocabulary presented a greater learning hurdle for the engineer and hardware types who have no interest in programming. There was much more to learn than was necessary for them. Of course, if they learned the language, they would have more power available.

Level 4.

With the demand for a more complete development system Tom Wempe developed his PADS system. At the same time, Tom Zimmer and collaborators, went through many versions of what became known as FPC.

FPC is a full Forth Development System with built in editor, debugger and many other features making it an outstanding package. However, the vocabulary has grown to nearly 2500 functions - A real learning challenge.

All of the source code is in text files. In fact, there are more source and help files in FPC than there are functions in fig-Forth. All possible speed is milked out of the processor. When added to the TCOM program small target compiled applications can be made.

The program and files are large. one needs a couple of megabytes of disk space available. To do all that it does, the space is needed. For the professional programmer, FPC is an excellent model. Ting has provided us the *F-PC Technical Reference Manual* and the *F-PC Users Manual.*

But for the engineer and hardware designer who is not interested in programming languages, the learning curve has increased. It may be more than he or she wants. However, with such a development system, Tom Zimmer has shown that hardware problems can be easily solved.

Level 5.

For many purposes, 32-bit values are overkill, but the are the coming processors. The segmented architecture of the Intel processors has led to severe limitations. Various schemes of memory management have been developed to work around some of those limitations. A 32-bit Forth system should have 32-bit stacks and address space. Some means should be found to deal efficiently with 16-bit addresses in segmented architecture.

F32, by Rick VanNorman solves some of these issues. He has created a full development package including text editor and assembler in about 900 words. This has been a outgrowth of his years of experience with Forth. He has placed the program in public domain and is available from several sources including Mountain View Press bulletin board.

His code is neat and well organized providing reasonable documentation for an experienced Forth programmer. However, there is as yet no printed documentation. I expect that there will be some before long. This implementation is not intended for a beginning Forth programmer.

Discussion.

These levels are oriented toward systems based on Intel processors. Most of them operate under a PC-DOS or MS-DOS environment. Other systems also make use of Forth. The Sun work Station is built with a Forth Monitor, and runs Unix. The Macintosh has available several Forth implementations. A Forth like Yerkes evolved out of NEON and runs on the Macintosh. These implementations are not for the beginner who wants a small language to get started. World wide there are still implementations of Forth.

Do not ignore the many professional Forth implementations. I consider these levels of Forth as models. Depending on ones interest and time, he can delve as deeply as he or she wishes.

For professional programmers, the professionally written implementations are well honed. They provide full development platforms. The big advantage of the professional systems is the many extensions and the level of documentation and support available from the respective companies.

The professional products include an appropriate tutorial. I have read the documentation and tried a number of the professional implementations. They are good. having an interest in implementation as well, I find the hidden proprietary nature of most of them to be frustrating. For professional programmer, these works offer a professional touch not found in the public domain implementations.

The problem then boils down to determining the programming level of the user and his applications. The user should select and master one tutorial or instructional manual. The order in each is different. Trying to learn from several at once will only confuse the student. Don't demand that the hardware designer and engineer be master computer scientists as well. Don't make the computer scientist work with minimal amateurish tools.

I think that there is a niche for Forth. The full blown development systems, whatever the language used, is overwhelming for many engineer and hardware designers. The interactive capability of Forth supplies these users a rapid means of progress. A small set of functions which meet his needs is sufficient.

All users are not equal.

A few final comments and updates. I checked the PC-SIG CDROM (release 11) and it contains MVP as released. The SIMTEL MSDOS CDROM has 8Forths. The 8 are, FPC 3.55, F83, FIFTH, FIG86, MINI4TH, PYGMY 1.3, Zen 1.10, and UniForth. I know that MINI, PYGMY, and Zen all are of Level 2 in size and work well. PYMGY is done by Frank Sergeant, TCJ's own PC/XT support person and a good starting Forth.

This paper is just as valid when considering our discussions of small-C. Small-C is somewhere between 0 and 1 in levels. What I am looking for in a TCJ language is something in a level 2. We have many Forth's starting with MVP Forth, but I am not sure about a C or Pascal in public domain that meets the level 2 parameters. Or do we? Bill Kibler with thanks to Glen Haydon.

Dr. S-100

By Herb R. Johnson

Regular Feature

Intermediate

Christmas Letters

"Dr. S-100's Christmas column" by Herb Johnson (c) Dec 1993

Introduction

Well, I got my Christmas wish early this year: another job! I'll refrain from telling all the exciting details, but it involves getting paid for doing work, a minimal requirement but a rarer event in the computer world. So, I'm left with less time for my interests in S-100; the Doctor has had to make extended rounds in the Computer Hospital these days, and hasn't kept up with his S-100 practice. So this month I won't play with BIOS's and disk controllers: maybe next time. (This gives you a chance to write me and offer encouragements or corrections: either would be appreciated!)

However, I have been in FidoNet correspondence with a new owner of a Morrow S-100 system. (Because I don't have permission from him, I'll leave him nameless.) He has owned several traditional CP/M systems, such as newer Morrows and Kaypros that are single-board systems, and has them running. This one arrived without docs, disks or even 8" drive cables and has had him stymied! But first, I'll run through the mail. One of my "patients" is very active in a strong interest of mine: astronomy!

Letters

Robert Grey of Chicago thanks me for my offer of assistance on his S-100 system and his unique application: "My Compupro Disk 1 [8" floppy disk controller] runs two Shugart 851's (a second generation: the first wore out). I kinda like the idea of being able to hang a 5" floppy drive on, and if a 1A would let me do just that, without patching anything or the chance of losing the functionality of my current driver, I'd like to do that.

Enclosed for your info is a newsletter on my small radio telescope - which is entirely automated under the Compupro and an IEEE-488 interface [also called HP-IB; it is an 8-bit interface somewhat like a SCSI], orchestrated by some 100 pages of FORTRAN code. Some of the graphs (enclosed) are done by the Diablo [daisy wheel printer] plotting package I wrote, which is very hard on the period (and hearing)!

I've got a few Compupro boards, plus a vast software library. I'd be interested in a faster CPU than the 5MHz 8085, if it's pretty close to plug and play...

Thanks again. The little green guys are elusive: [Robert is engaged in a SETI (search for extra-terrestrial intelligence) project spanning ten years] it appears that I'll have to keep looking for a long time, and S-100 hardware might just carry me through!"

Robert called me earlier to tell me that he had zapped his Compupro boards with AC current, which he uses to operate his 12-foot steerable dish antenna. He is looking at a commonly-postulated frequency of 1420 MHz, or a wavelength of 23 centimeters. This is a galactic "quiet zone" in the radio spectrum, and the frequency of atomic emissions of hydroxyl (OH) molecules commonly called the "waterhole" frequency. This SETI search is part of a long standing, legitimate effort by scientists and amateurs to find deliberate radio signals from another civilization (even if Congress disagrees about funding it).

I sent him a few cards, working and semi-working, for him to consider. He sent me a descriptive brochure on his work to date, and its relation to another SETI program I participated in at Ohio State some years ago which is still in progress. Maybe I'll encourage Robert to write an article for some issue!

Albert Wojda of Lyons, IL was kind enough to send me a list of his cards, available with docs and disks. They include Compupro, Fulcrum (IMSAI), Godbout and Jade cards. **C. M. Kotlan** of Gold Hill, Oregon has an XOR S-100 system for sale, with two double sided 8-inch drives; and some Compupro equipment.

Stephen Griswold of CT came to the Doctor's aid recently: I sent out a request on the FidoNet (network of BBS systems) CP/M Tech echo (message area) for a ROM card for the Processor Tech SOL system. I have two systems, but only one ROM card. Stephen sent one through the mail, less ROMS and address chip as his message had described. Thanks and a free Bus pass to you!

A poignant note from the UK

Emmanuel Roche, from somewhere in France, sent me a copy of **The Windsor Bulletin Board User's Group Disk Library News**. I presume this CP/M group is in Windsor in England, as there is no address within it. Anyone have a clue? The newsletter suggest that Digital Research placed a "lot of GSX-86 software into the public domain...we intend to write to [DR to release the GSX-80 material]". I remember this as a graphics standard from around the time of CP/M 3.0. The editor notes in international news that, after a poor response from correspondents in Germany and the US that "I myself shall not abandon CP/M because I see no advantage in doing so, and I am not subjected to commercial pressures to do so, but there seems little evidence even in the States, with its much larger population, that that outlook is a common one.

"I've fallen over a S-100 system and I can't get it up!"

In his FidoNet correspondence, this new S-100 owner has tried to identify what he has to people who might help him, and he has been frustrated for a number of reasons. I thought I would share some of his issues and his activities with my readers and provide you with some guidance about how S-100 systems are different from the "classic" luggable and non-bus-based desktop CP/M systems, and how you can re-activate them.

1) "S-100 systems have a reputation for reliability **and** for unreliability."

That is correct! S-100 boards were very popular from the mid-1970's to the early 1980's. In those days chips with complex functions or a lot of memory were very expensive; and even small-scale logic chips (chips with a few logic gates or register bits) were a dollar or two each. Also, the cost of printed circuit boards was significant. The virtue of the S-100 world was that as a manufacturer you did not have to offer the whole computer; just one specialized card! Consequently, every one and their cousin designed, built and sold cards.

The amazing thing is that **most of them worked!** The chips of the times generally operated much faster than the processors and memories of the time, yet were slow enough that sloppy board layouts and marginal designs still operated - usually. And, the S-100 specification (mostly an 8080/Z-80 timing specification) was fairly tolerant of timing problems.

The biggest problem of these early designs was **heat sensitivity**. For example, the IMSAI front panel relies on what is called "one-shots" or monostable flipflops. These were essentially timer chips (such as the 74121 and 74123) that created a pulse of fixed width when triggered by a pulse or the leading edge (start) of a digital event (the start of a read cycle, for instance). The monostable used as a pulse-width reference the charge of a capacitor through a resistor, commonly called a "R-C circuit". Capacitance and resistance can change with heat, as can the "threshold" value of charge (voltage) required to complete the pulse cycle. For critical timing of events this could cause intermittent or continual failure until the monostable "cooled down". With all the boards and chips in a S-100 system, not to mention the **hot voltage regulators**, a system could get quite warm in ten or fifteen minutes!

The current problem with old S-100 systems is **age**. Components, most notably capacitors, shift in value or even fail over several years. A more spectacular failure than "bad timing" occurs with the **tantalum capacitors** used to filter the five-volt power lines on boards. Given that an S-100 system eats several amps of current at five volts, and that each board has its own set of voltage regulators, the S-100 supplies provide 8 or even more volts at **10 or 15 amps, or higher**. Tantalum capacitors tend to fail by shorting: a bad idea across a power supply line! Result: a firecracker "pop" of the capacitor, with smoke and even flames! While not dangerous - the whole board is not likely to ignite - is certainly is startling.

So where does the reliability come in? A reasonably designed card would not include monostables and other non-digital timing and delay circuits. It would use chips intended for operation at the speeds recommended for the card, or even slower. Today's experienced users can try to obtain reliability by substituting slower CPU cards, or by using a Z80 CPU card instead of an 8080 or 8085 (the Z80 has a more "forgiving" read cycle). Many people have systems with 10-year-old cards that run well. Newer S-100 bus boards included a circuit called **active termination**, which fed a regulated voltage to each active bus pin, and which also included a resistor from each pin to ground and to +5 volts. Older systems can be adapted with an active termination card, or even a "passive" terminator with resistors only.

Other things a current user can do to improve reliability is to use newer memory cards, particularly static memory cards. Dynamic memory requires more chips and more design effort than static. In the old days, you could typically get four times more memory per dollar by going dynamic. For those of you not familiar with specific memory chips, most dynamic RAM chips are in 16 or 18-pin packages (except for the very oldest ones of 1976 or so) while most static RAM chips were in wider packs with 24 or 28 pins, similar to ROM chip packages. The exceptions in static RAM are the 2102 and the 4114's, which are 16-pin 1K and 4K chips, respectively. Another advantage of newer cards is that they contain more memory: using fewer cards means less heat in your S-100 box.

2) "What boards do I need to make this system run?"

The S-100 architecture was started by MITS with the Altair; almost immediately it was modified by IMSAI, and slightly more mods later by Cromemco. By the early 1980's there was a standard developed for a S-100 bus with more address and data lines and more features called IEEE-696. This standard was followed by most manufacturers after the early 80's.

I bring this history up to underscore some problems in plugging "any" board into a S-100 box. Given the changes in bus standards, and given the design problems I mentioned earlier, my recommendation in general is to **stick with one manufacturer.** The exception is the early days of S-100, the Altair, IMSAI, early Compupro and Cromemco cards and other boards built in the 1970's: most of those, but not all, are at least bus compatible.

This is generally not a big issue: most newcomers to S-100 start with a system already "integrated" and (at one time) running. More likely, they may not have docs or disks: "those were in another box, I don't know where it went". This leads to the next question:

3) "What are the secret instructions to make this system operate?"

Years of Kaypro's and Osborne's, followed by a decade of IBM-PC's and Macs, have generated a breed of computer user that expects computers to operate in **only one way**. Thus the growth of "wizard" books (to learn the secrets) and "dummy" books (if you don't know the secrets). Don't get me wrong: standards allow anyone who knows how to operate one IBM-PC to operate another PC anywhere else...more or less. However, S-100 systems are not all alike, or even identical to each other, unless they have the same cards by the same manufacturer, with the same ROM's and BIOS's.

Let me be clear on that last point: S-100 systems are NOT like Kaypros, or Osborne, or Zorbas, etc. The latter are single-board systems with only ONE way to operate. Your Zorba will run just like my Zorba. This makes them easier to compare, to diagnose and to learn to run. They may not be "easy" to repair, but you can find the problem fast. S-100 systems are both rarer and more unique.

4) "So how can I repair or operate my ****** S-100 system? The terminal shows a message ("insert disk" or "Morrow V2.99" or "?" or some such prompt) but it doesn't boot from any of my disks (usually from another system)."

I'm sorry that I don't know all the cards in your system. So, I can't say what is and is not working PRECISELY. Given that your terminal displays some information from the system on power up, obviously the CPU card must be working. Does the CPU card have a ROM on it? Does it have RAM on it? (Refer to the previous chip description for help.) If the RAM memory or ROM are on separate cards then the bus is probably ok too. So, you have a system with a ROM monitor and some working RAM. If you know how to fill memory with a particular test value via the monitor, you can do this selectively to see if parts of memory are overwriting each other, or to see if memory values are "stable" over time.

To get some idea of the monitor commands, try each letter of the alphabet, capitalized, followed by the number 1000. Also, try putting a space between the letter and the number. Also, try a pair of numbers, separated by a space, hyphen, comma, or colon. If the disk drives are connected, look for a drive light or listen for the sound of a drive's head engaging on the diskette, or the sound of the heads moving. (Put a diskette in the drive for these tests, one that you can afford to write over, and preferably formatted in some fashion.)

If the monitor supports "disk read" and "disk write" commands, then you could take a formatted diskette and try to dump sectors off of it. Disks from different machines may have different sector sizes, and your machine may only accept one size. In any case, it will be easier to get the floppy going than the hard drive (if it has one), because you can insert formatted diskettes into it: hard drives are not "universally formatted" and older hard disk controllers are harder to deal with (because they were designed in many ways in the old days).

If you are familiar with IC chips, and which ones do what sort of major things (like the 40-pin CPU chips, UART chips, and the smaller chips for ROM and RAM) you can get pretty far without docs. If you have an oscilloscope and knew about the chips I mentioned, you could check certain pins (address and more importantly "chip enable" pins) for receiving addressing signals to see if they are being accessed. The bottom line is that your ability to diagnose this system depends on either being able to read the docs that you can get, or to figure out from the chips in use what each board does.

5) I think this system requires a hard-sectored diskette. How can I determine this from the disk drive?

(I'm surprised how often this question comes up.) Briefly, you can't: Hard sectoring or soft sectoring is a function of the CONTROLLER, not the drive. In the old days, some computers used to have their diskettes marked with a hole for each sector; as well as a hole for the "index", or start of a track. They were all readable with the same sensor, a LED and a optical diode or transistor with the disk in between. The index hole is between two sector holes. So there is no difference between a "soft sectored" drive and a "hard sectored" drive.

6) I've tried all the boot disks I have, and none of them work!

No two manufacturer's CP/M systems boot from the same disks! Each system has its own code for bootup and for the CP/M BIOS usually contained on the "boot tracks". (Read my article from last issue for details.) You need specific boot disks from the system's manufacturer (or at least the disk controller's manufacturer) to boot up; or you need to be able to recreate them from "scratch". The latter is a challenge for some, a dead end for others.

The good news is that there are a few people who have collections of disks, or whom own the system you are looking at. The most important thing to do is to document what you have, so that others can figure out what you have.

7) "OK, how do I figure out what I have?"

Get a pad of paper. Open up the system, pull out one card, noting all the cables and their locations. Write down all the information you see on the card, most notably the manufacturer and name of the card. Note any large chips, and note any chips with identical numbers that are placed in an obvious grid pattern. Write down the chips numbers and names as you see them. Also note the cable

Conclusion continued on page 50.

The Computer Journal

Back Issues

Sales limited to supplies in stock.

Volume Number 1:

- **Issues 1 to 9**
- **Serial Interfacing and Modem transfers**
- **Floppy disk formats, Print spooler.**
- **Adding 8087 Math Chip, Fiber optics**
- **S-100 HI-RES graphics.**
- **Controlling DC motors, Multi-user column.**
- **VIC-20 EPROM Programmer, CP/M 3.0.**
- **CP/M user functions and Integration.**

Volume Number 2:

- **Issues 10 to 19**
- **Forth tutorial and Write Your Own.**
- **68008 CPU for S-100.**
- **RPM vs CP/M, BIOS Enhancements.**
- **Poor Man's Distributed Processing.**
- **Controlling Apple Stepper Motors.**
- **Facsimile Pictures on a Micro.**
- **Memory Mapped I/O on a ZX81.**

Volume Number 3:

- **Issues 20 to 25**
- Designing an 8035 SBC
- Using Apple Graphics from CP/M
- Soldering & Other Strange Tales
- Build an S-100 Floppy Disk Controller: WD2797 Controller for CP/M 68K
- Extending Turbo Pascal: series
- Unsoldering: The Arcane Art
- Analog Data Acquisition & Control: Connecting Your Computer to the Real World
- Programming the 8035 SBC
- NEW-DOS: series
- Variability in the BDS C Standard Library
- The SCSI Interface: series
- Using Turbo Pascal ISAM Files
- The Ampro Little Board Column: series
- C Column: series
- The Z Column: series
- The SCSI Interface: Introduction to SCSI
- Editing the CP/M Operating System
- INDEXER: Turbo Pascal Program to Create an Index
- Selecting & Building a System
- Introduction to Assemble Code for CP/M
- Ampro 186 Column
- ZTime-1: A Real Time Clock for the Ampro Z-80 Little Board

Issue Number 26:

- Bus Systems: Selecting a System Bus
- Using the SB180 Real Time Clock
- The SCSI Interface: Software for the SCSI Adapter
- Inside Ampro Computers
- NEW-DOS: The CCP Commands (continued)
- ZSIG Corner
- Affordable C Compilers
- Concurrent Multitasking: A Review of DoubleDOS

Issue Number 27:

- 68000 TinyGiant: Hawthorne's Low Cost 16-bit SBC and Operating System
- The Art of Source Code Generation: Disassembling Z-80 Software
- Feedback Control System Analysis: Using Root Locus Analysis & Feedback Loop Compensation
- The C Column: A Graphics Primitive Package
- The Hitachi HD64180: New Life for 8-bit Systems
- ZSIG Corner: Command Line Generators and Aliases
- A Tutor Program in Forth: Writing a Forth Tutor in Forth
- Disk Parameters: Modifying the CP/M Disk Parameter Block for Foreign Disk Formats

Issue Number 28:

- Starting Your Own BBS
- Build an A/D Converter for the Ampro Little Board
- HD64180: Setting the Wait States & RAM Refresh using PRT & DMA
- Using SCSI for Real Time Control
- Open Letter to STD Bus Manufacturers
- Patching Turbo Pascal
- Choosing a Language for Machine Control

Issue Number 29:

- Better Software Filter Design
- MDISK: Adding a 1 Meg RAM Disk to Ampro Little Board, Part 1
- Using the Hitachi hd64180: Embedded Processor Design
- 68000: Why use a new OS and the 68000?
- Detecting the 8087 Math Chip
- Floppy Disk Track Structure
- The ZCPR3 Corner

Issue Number 30:

- Double Density Floppy Controller
- ZCPR3 IOP for the Ampro Little Board
- 3200 Hackers' Language
- MDISK: Adding a 1 Meg RAM Disk to Ampro Little Board, Part 2
- Non-Preemptive Multitasking
- Software Timers for the 68000
- Lilliput Z-Node
- The ZCPR3 Corner
- The CP/M Corner

Issue Number 31:

- Using SCSI for Generalized I/O
- Communicating with Floppy Disks: Disk Parameters & their variations
- XBIOS: A Replacement BIOS for the SB180
- K-OS ONE and the SAGE: Demystifying Operating Systems
- Remote: Designing a Remote System Program
- The ZCPR3 Corner: ARUNZ Documentation

Issue Number 32:

- Language Development: Automatic Generation of Parsers for Interactive Systems
- Designing Operating Systems: A ROM based OS for the Z81
- Advanced CP/M: Boosting Performance
- Systematic Elimination of MS-DOS Files: Part 1, Deleting Root Directories & an In-Depth Look at the FCB
- WordStar 4.0 on Generic MS-DOS Systems: Patching for ASCII Terminal Based Systems
- K-OS ONE and the SAGE: System Layout and Hardware Configuration
- The ZCPR3 Comer: NZCOM and ZCPR34

Issue Number 33:

- Data File Conversion: Writing a Filter to Convert Foreign File Formats
- Advanced CP/M: ZCPR3PLUS & How to Write Self Relocating Code
- DataBase: The First in a Series on Data Bases and Information Processing
- SCSI for the S-100 Bus: Another Example of SCSI's Versatility
- A Mouse on any Hardware: Implementing the Mouse on a Z80 System
- Systematic Elimination of MS-DOS Files: Part 2, Subdirectories & Extended DOS Services
- ZCPR3 Corner: ARUNZ Shells & Patching WordStar 4.0

Issue Number 34:

- Developing a File Encryption System.
- Database: A continuation of the data base primer series.
- A Simple Multitasking Executive: Designing an embedded controller multitasking executive.
- ZCPR3: Relocatable code, PRL files, ZCPR34, and Type 4 programs.
- New Microcontrollers Have Smarts: Chips with BASIC or Forth in ROM are easy to program.
- Advanced CP/M: Operating system extensions to BDOS and BIOS, RSXs for CP/M 2.2.
- Macintosh Data File Conversion in Turbo Pascal.

Issue Number 35:

- All This & Modula-2: A Pascal-like alternative with scope and parameter passing.
- A Short Course in Source Code Generation: Disassembling 8088 software to produce modifiable assem. source code.
- Real Computing: The NS32032.
- S-100: EPROM Burner project for S-100 hardware hackers.
- Advanced CP/M: An up-to-date DOS, plus details on file structure and formats.
- REL-Style Assembly Language for CP/M and Z-System. Part 1: Selecting your assembler, linker and debugger.

Issue Number 36:

- Information Engineering: Introduction.
- Modula-2: A list of reference books.
- Temperature Measurement & Control: Agricultural computer application.
- ZCPR3 Corner: Z-Nodes, Z-Plan, Amstrand computer, and ZFILE.
- Real Computing: NS32032 hardware for experimenter, CPUs in series, software options.
- SPRINT: A review.
- REL-Style Assembly Language for CP/M & ZSystems, part 2.
- Advanced CP/M: Environmental programming.

Issue Number 37:

- C Pointers, Arrays & Structures Made Easier: Part 1, Pointers.
- ZCPR3 Corner: Z-Nodes, patching for NZCOM, ZFILER.
- Information Engineering: Basic Concepts: fields, field definition, client worksheets.
- Shells: Using ZCPR3 named shell variables to store date variables.
- Resident Programs: A detailed look at TSRs & how they can lead to chaos.
- Advanced CP/M: Raw and cooked console I/O.
- Real Computing: The NS 32000.
- ZSDOS: Anatomy of an Operating System: Part 1.

Issue Number 38:

- C Math: Handling Dollars and Cents With C.
- Advanced CP/M: Batch Processing and a New ZEX.
- C Pointers, Arrays & Structures Made Easier: Part 2, Arrays.
- The Z-System Corner: Shells and ZEX, new Z-Node Central, system security under Z-Systems.
- Information Engineering: The portable Information Age.
- Computer Aided Publishing: Introduction to publishing and Desk Top Publishing.
- Shells: ZEX and hard disk backups.
- Real Computing: The National Semiconductor NS320XX.
- ZSDOS: Anatomy of an Operating System, Part 2.

Issue Number 39:

- Programming for Performance: Assembly Language techniques.
- Computer Aided Publishing: The Hewlett Packard LaserJet.
- The Z-System Corner: System enhancements with NZCOM.
- Generating LaserJet Fonts: A review of Digi-Fonts.
- Advanced CP/M: Making old programs Z-System aware.
- C Pointers, Arrays & Structures Made Easier: Part 3: Structures.
- Shells: Using ARUNZ alias with ZCAL.
- Real Computing: The National Semiconductor NS320XX.

Issue Number 40:

- Programming the LaserJet: Using the escape codes.
- Beginning Forth Column: Introduction.
- Advanced Forth Column: Variant Records and Modules.
- LINKPRL: Generating the bit maps for PRL files from a REL file.
- WordTech's dBXL: Writing your own custom designed business program.
- Advanced CP/M: ZEX 5.0×The machine and the language.
- Programming for Performance: Assembly language techniques.
- Programming Input/Output With C: Keyboard and screen functions.
- The Z-System Corner: Remote access systems and BDS C.
- Real Computing: The NS320XX

Issue Number 41:

- Forth Column: ADTs, Object Oriented Concepts.
- Improving the Ampro LB: Overcoming the 88Mb hard drive limit.
- How to add Data Structures in Forth
- Advanced CP/M: CP/M is hacker's haven, and Z-System Command Scheduler.
- The Z-System Corner: Extended Multiple Command Line, and aliases.
- Programming disk and printer functions with C.
- LINKPRL: Making RSXes easy.
- SCOPY: Copying a series of unrelated files.

Issue Number 42:

- Dynamic Memory Allocation: Allocating memory at runtime with examples in Forth.
- Using BYE with NZCOM.
- C and the MS-DOS Screen Character Attributes.
- Forth Column: Lists and object oriented Forth.
- The Z-System Corner: Genie, BDS Z and Z-System Fundamentals.
- 68705 Embedded Controller Application: An example of a single-chip microcontroller application.
- Advanced CP/M: PluPerfect Writer and using BDS C with REL files.
- Real Computing: The NS 32000.

Issue Number 43:

- Standardize Your Floppy Disk Drives.
- A New History Shell for ZSystem.
- Heath's HDOS, Then and Now.
- The ZSystem Corner: Software update service, and customizing NZCOM.
- Graphics Programming With C: Graphics routines for the IBM PC, and the Turbo C graphics library.
- Lazy Evaluation: End the evaluation as soon as the result is known.
- S-100: There's still life in the old bus.
- Advanced CP/M: Passing parameters, and complex error recovery.
- Real Computing: The NS32000.

Issue Number 44:

- Animation with Turbo C Part 1: The Basic Tools.
- Multitasking in Forth: New Micros F68FC11 and Max Forth.
- Mysteries of PC Floppy Disks Revealed: FM, MFM, and the twisted cable.
- DosDisk: MS-DOS disk format emulator for CP/M.
- Advanced CP/M: ZMATE and using lookup and dispatch for passing parameters.
- Real Computing: The NS32000.
- Forth Column: Handling Strings.
- Z-System Corner: MEX and telecommunications.

Issue Number 45:

- Embedded Systems for the Tenderfoot:

The Computer Journal Back Issues

Getting started with the 8031.
- The Z-System Corner: Using scripts with MEX.
- The Z-System and Turbo Pascal: Patching TURBO.COM to access the Z-System.
- Embedded Applications: Designing a Z80 RS-232 communications gateway, part 1.
- Advanced CP/M: String searches and tuning Jetfind.
- Animation with Turbo C: Part 2, screen interactions.
- Real Computing: The NS32000.
- The Computer Corner.

Issue Number 46:
- Build a Long Distance Printer Driver. Using the 8031's built-in UART for serial communications.
- Foundational Modules in Modula 2.
- The Z-System Corner: Patching The Word Plus spell checker, and the ZMATE macro text editor.
- Animation with Turbo C: Text in the graphics mode.
- Z80 Communications Gateway: Prototyping, Counter/Timers, and using the Z80 CTC.

Issue Number 47:
- Controlling Stepper Motors with the 68HC11F
- Z-System Corner: ZMATE Macro Language
- Using 8031 Interrupts
- T-1: What it is & Why You Need to Know
- ZCPR3 & Modula, Too
- Tips on Using LCDs: Interfacing to the 68HC705
- Real Computing: Debugging, NS32 Multi-tasking & Distributed Systems
- Long Distance Printer Driver: correction
- ROBO-SOG 90
- The Computer Corner

Issue Number 48:
- Fast Math Using Logarithms
- Forth and Forth Assembler
- Modula-2 and the TCAP
- Adding a Bernoulli Drive to a CP/M Computer (Building a SCSI Interface)
- Review of BDS "Z"
- PMATE/ZMATE Macros, Pt. 1
- Real Computing
- Z-System Corner: Patching MEX-Plus and TheWord, Using ZEX
- Z-Best Software
- The Computer Corner

Issue Number 49:
- Computer Network Power Protection
- Floppy Disk Alignment w/RTXEB, Pt. 1
- Motor Control with the F68HC11
- Controlling Home Heating & Lighting, Pt. 1
- Getting Started in Assembly Language
- LAN Basics
- PMATE/ZMATE Macros, Pt. 2
- Real Computing
- Z-System Corner
- Z-Best Software
- The Computer Corner

Issue Number 50:
- Offload a System CPU with the Z181
- Floppy Disk Alignment w/RTXEB, Pt. 2
- Motor Control with the F68HC11
- Modula-2 and the Command Line
- Controlling Home Heating & Lighting, Pt. 2
- Getting Started in Assembly Language Pt 2
- Local Area Networks
- Using the ZCPR3 IOP
- PMATE/ZMATE Macros, Pt. 3
- Z-System Corner, PCED
- Z-Best Software
- Real Computing, 32FX16, Caches
- The Computer Corner

Issue Number 51:
- Introducing the YASBEC
- Floppy Disk Alignment w/RTXEB, Pt 3
- High Speed Modems on Eight Bit Systems
- A Z8 Talker and Host
- Local Area Networks--Ethernet
- UNIX Connectivity on the Cheap
- PC Hard Disk Partition Table
- A Short Introduction to Forth
- Stepped Inference as a Technique for Intelligent Real-Time Embedded Control
- Real Computing, the 32CG160, Swordfish, DOS Command Processor
- PMATE/ZMATE Macros
- Z-System Corner, The Trenton Festival
- Z-Best Software, the Z3HELP System
- The Computer Corner

Issue Number 52:
- YASBEC, The Hardware
- An Arbitrary Waveform Generator, Pt. 1
- B.Y.O. Assembler...in Forth
- Getting Started in Assembly Language, Pt. 3
- The NZCOM IOP
- Servos and the F68HC11
- Z-System Corner, Programming for Compatibility
- Z-Best Software
- Real Computing, X10 Revisited
- PMATE/ZMATE Macros
- Controlling Home Heating & Lighting, Pt. 3
- The CPU280, A High Performance Single-Board Computer
- The Computer Corner

Issue Number 53:
- The CPU280
- Local Area Networks
- Am Arbitrary Waveform Generator
- Real Computing
- Zed Fest '91
- Z-System Corner
- Getting Started in Assembly Language
- The NZCOM IOP
- Z-BEST Software
- The Computer Corner

Issue Number 54:
- Z-System Corner
- B.Y.O. Assembler
- Local Area Networks
- Advanced CP/M
- ZCPR on a 16-Bit Intel Platform
- Real Computing
- Interrupts and the Z80
- 8 MHZ on a Ampro
- Hardware Heavenn
- What Zilog never told you about the Super8
- An Arbitary Waveform Generator
- The Development of TDOS
- The Computer Corner

Issue Number 55:
- Fuzzilogy 101
- The Cyclic Redundancy Check in Forth
- The Internetwork Protocol (IP)
- Z-System Corner
- Hardware Heaven
- Real Computing
- Remapping Disk Drives through the Virtual BIOS
- The Bumbling Mathmatician
- YASMEM
- Z-BEST Software
- The Computer Corner

Issue Number 56:
- TCJ - The Next Ten Years
- Input Expansion for 8031
- Connecting IDE Drives to 8-Bit Systems
- Real Computing
- 8 Queens in Forth
- Z-System Corner
- Kaypro-84 Direct File Transfers
- Analog Signal Generation
- The Computer Corner

Issue Number 57:
- Home Automation with X10
- File Transfer Protocols
- MDISK at 8 MHZ.
- Real Computing
- Shell Sort in Forth
- Z-System Corner
- Introduction to Forth
- DR. S-100
- Z AT Last!
- The Computer Corner

Issue Number 58:
- Multitasking Forth
- Computing Timer Values
- Affordable Development Tools
- Real Computing
- Z-System Corner
- Mr. Kaypro
- DR. S-100
- The Computer Corner

Issue Number 59:
- Moving Forth
- Center Fold IMSAI MPU-A
- Developing Forth Applications
- Real Computing
- Z-System Corner
- Mr. Kaypro Review
- DR. S-100
- The Computer Corner

Issue Number 60:
- Moving Forth Part II
- Center Fold IMSAI CPA
- Four for Forth
- Real Computing
- Debugging Forth
- Support Groups for Classics
- Z-System Corner
- Mr. Kaypro Review
- DR. S-100
- The Computer Corner

Issue Number 61:
- Multiprocessing 6809 part I
- Center Fold XEROX 820
- Quality Control
- Real Computing
- Support Groups for Classics
- Z-System Corner
- Operating Systems - CP/M
- Mr. Kaypro 5MHZ
- The Computer Corner

Issue Number 62:
- SCSI EPROM Programmer
- Center Fold XEROX 820
- DR S-100
- Real Computing
- Moving Forth part III
- Z-System Corner
- Programming the 6526 CIA
- Reminiscing and Musings
- Modem Scripts

Issue Number 63:
- SCSI EPROM Programmer part II
- Center Fold XEROX 820
- DR S-100
- Real Computing
- Multiprocessing Part II
- Z-System Corner
- 6809 Operating Systems
- Reminiscing and Musings
- IDE Drives Part II

Issue Number 64:
- Small-C?
- Center Fold last XEROX 820
- DR S-100
- Real Computing
- Moving Forth Part IV
- Z-System Corner
- Small Systems
- Mr. Kaypro
- IDE Drives Part III

SPECIAL DISCOUNT

15% on cost of Back Issues when buying from 1 to Current Issue.
10% on cost of Back Issues when buying 20 or more issues.

	U.S.	Canada/Mexico		Europe/Other	
Subscriptions (CA not taxable)		(Surface)	(Air)	(Surface)	(Air)
1year (6 issues)	$24.00	$32.00	$34.00	$34.00	$44.00
2 years (12 issues)	$44.00	$60.00	$64.00	$64.00	$84.00
Back Issues (CA tax)	add these shipping costs for each issue ordered				
Bound Volumes $20.00 ea	+$3.00	+$3.50	+$6.50	+$4.00	+$17.00
#20 thru #43 are $3.00 ea.	+$1.00	+$1.00	+$1.25	+$1.50	+$2.50
#44 and up are $4.00ea.	+$1.25	+$1.25	+$1.75	+$2.00	+$3.50
Software Disks (CA tax)	add these shipping costs for each 3 disks ordered				
MicroC Disks are $6.00ea	+$1.00	+$1.00	+$1.25	+$1.50	+$2.50

Items: ______________ Back Issues Total ________

______________ MicroC Disks Total ________

California state Residents add 7.25% Sales TAX ________

Subscription Total ________

Total Enclosed ________

Name: ______________________

Address: ______________________

____

____

____

Credit Card # _____-_____-_____-_____ exp ___/___

Payment is accepted by check, money order, or Credit Card (M/C, VISA, CarteBlanche, Diners Club). Checks must be in US funds, drawn on a US bank. Credit Card orders can call 1(800) 424-8825.

TCJ The Computer Journal

P.O. Box 535, Lincoln, CA 95648-0535
Phone (916) 645-1670

TCJ CLASSIFIED

WANTED: Back issues of TCJ. Want the actual copies, no reprints. Looking for 1, 2, 5 to 17, 19, 24, and 26. Willing to pay reasonable amounts, including shipping. Keith Tucholski, (717) 929-2658, or CompuServe 73441,265.

Needed: 200 Timex/Sinclair computers or compatibles. Xavier Ornelas, Bocoyna 302, 31600 Anahuac, Chih. Mexico.

Needed: Z80 replacement for Microprofessor Instructional Z80 systems. Used in classroom courses at college level. Dr, Mauro Caputi, Hopstra Engr. Dept., 133 Hopstra Univ., Hempstead NY 11550. (516) 463-5549.

Needed: Boot Disk or help in restoring IMS Corp. 518-W12-10H60, has 10 meg HD. Had boot code stored in battery backedup memory - battery died and will not boot now. John Edgecombe, 9546 Los Palos Rd. Atascadero, CA 93422.

For Sale: ***Collector's Guide to Personal Computers and Pocket Calculators***. Prices and illustations included. 36 pages, $14.95 plus $2.00 shipping. Fred Hartfield, Box 52466, New Orleans, LA 70152. Digitial Cottage BBS (504) 897-5514, help, support, sales, of old systems.

Avaialble: *MicroC* Kaypro Disks, *MicroC* Back Issues and legal copies of CP/M, Borland Software (Turbo Pascal, DBase II, and more). Many bootable CP/M disk formats avaialble. Disk copying, most formats inculding Apple CP/M. Manuals and more! Lambda Software Publishing, 149 West Hilliard Lane, Eugene, OR 97404-3057, (503) 688-3563.

NEW CLASSIFIED RATES!

NOW $5.00 PER LISTING!

TCJ Classified ads are on a prepaid basis only. The cost is $5.00 per ad entry. Support wanted is a free service to subscribers who need to find old or missing documentation or software. Please limit your requests to one type of system.

Commercial Advertising Rates:

Size	Once	4+
Full	$120	$90
1/2 Page	$75	$60
1/3 Page	$60	$45
1/4 Page	$50	$40
Market Place	$25	$100/yr

Send your items to:

The Computer Journal
P.O. Box 535
Lincoln, CA 95648-0535

Dr. S-100 Continued from page 47.

connections: where do the cables go, how many pins are on the cables, etc.

Look for chips with these numbers: 1488 or 1489, or 74188 and 74189, or similar numbers. These chips are used for the serial interface, and should be near a 16-pin or 26-pin connector (pin counts can vary). Not all serial interfaces use these chips, but most do.

Then put the card back, and get the next one. Repeat for all boards, including the mother or bus board (don't remove it!). Now you can search for someone with identical cards or a very similar system with whom you can swap docs, cards and maybe get a boot disk. At the least, you can send your list around and someone can read it and at least tell you what you have!

The bottom line, I guess, is that S-100 systems are designed to be tinkered with; if you don't enjoy "tinkering" at the chip level, and you don't have docs or don't know the chips of the 1970's and 1980's, they are probably not "your" kind of machine. Unless you want to learn that kind of stuff, or you need a machine to do unusual things, like operating a radio telescope!

References

Robert H Gray, Small SETI Radio Telescope, 3071 Palmer Square, Chicago IL 60647.

Albert Woida, 4023 Anna Ave, Lyons IL 60534-1011. (708) 442-9261 evenings.

C M Kotlan, PO Box 134, Gold Hill Oregon 97525.

Stephen Griswold, C/O the Fidonet CPM-TECH echo. Contact your local or regional BBS sysops and ask about carrying the FidoNet message areas.

The Computer Corner

By Bill Kibler

Regular Feature

Editorial Comment

OS/2 Plus

Welcome to the ever changing world of computer support. Seems like everybody has changed jobs lately, including me. Since I have left the OS/2 world, I figure I had better cover what I learned before I forget it.

My overall feelings about OS/2 before working on it, was not very clear or good. I was pretty skeptical of the whole product. I knew my old company had major programming problems using it and not too many people were giving it much praise.

The most recent release, Version 2.1, however seemed to change many minds over it's merits. I was moved from 68000 assembly to C on OS/2 at work and thus was able to get my own ideas about it's merits.After several months of programming and sitting in front of it daily for hours, I found it OK. Since OK is pretty vague, we need some experiences to help qualify that statement.

First off you will find the books almost non existent. Sacramento's largest book store has only one foot of shelf space for OS/2, while the just released NT has 10 times that already. The counter to this problem is, you might not need any books. OS/2 has plenty of on-line help files, works closely with and like Windows, and once installed usually works without problems.

Two good books to get are "OS/2 2.1 UNLEASHED" and "Real World Programming for OS/2 2.1." These two books are current, large, comprehensive, and very handy. That and what is on line may be all you ever need. The reason you might want the unleashed book is handling problems with initial installation. When installing OS/2 it can go either faultlessly or be a mess. Absolute reading of Appendix G is a must. That section explains which BIOS version you MUST have in order for OS/2 to work. To use IDE drives, the BIOS must be newer than 040990 (April of 1990). Before this date the IDE interfaces were not correct and will crash OS/2.If you run SCSI drives and your keyboard is ok, even some older ROMs will work. The keyboard prefix is typically "F" or greater (last value of ROM serial code "-F".) This is based on the type and nature of the Keyboard controller interface software.

Should you consider any variation to a simple installation, read the appendixes, in this case all of them. Some mention in the Installation Guide to looking in the appendix is made as you read the early chapters, but you probably will underestimate their importance. Don't!

Having hopefully guided you through some pitfalls, you should find it easy to use after this. I used my DOS editor without problems, but many of the DOS utilities become useless from the OS2 command line. You can run DOS programs from the OS2 command line, and OS2 will automatically switch to DOS command line, run the program, and then return you to OS/2, usually long before you could read the results of your efforts. This can be easily resolved by just running DOS and OS/2 command windows instead of dropping out of PM.

Oh, PM, that is OS/2's window like control or main screen. Everything runs from it so to speak. I hope you are getting the picture, OS/2 is much more complex that DOS or CP/M. There will be lots of new terms and concepts to be learned. Those new items will give you the ability to have many programs running at the same time.

At this point I want to move on to other topics and request that Rick Rodman or a reader's article follow up on OS/2, as many pages would be needed to guide you through OS/2.

The New Job

My previous commute was very long and each day lasted forever because of it. I also really didn't like OS/2 and especially the C programming. The project was a quagmire of problems. Some of those problems were due to OS/2 1.0 having so many bugs in it. The early programmers programmed around many of those early bugs.

When version 2.1 arrived, the project had become so monstrous, removing and simplifying the code was no longer possible. Those early needed fixes had now become part of the structure and not just removable fixes.

All those problems plus a company in Chapter 11 had me looking for something different. I had already decided I didn't want to be on the leading edge of software development. It seemed to me that a lot of what was going on, was keeping up with the latest bug that was created when Microsoft released a new version of DOS or Windows, and now IBM and OS/2. It seemed like we never really worked on the programs structure, but worked instead on forced changes from the OS bug fixes.

I wanted out of that rat race and found it. My new company provides security systems for prisons. It's just what I wanted. We use computers, but mostly as embedded work horses. Overall my job will be doing ladder logic on PLC's (programmable logic controllers), some assembly language on 8048/51's, a little PLC BASIC programming, some hardware design work, and STD bus upgrading. Since I like having fun at work, not working on OS/2 and C in my new job, make having fun possible again.

PCW 8256

After setting up Helmut's article, I stopped by one of the local used vendors. He is now selling used, complete XT systems with monitors for under $100! Most have hard drives, although small (10 or 20 Meg) a working hard drive none the less. He also has complete Z-100's with hard drives for $45 which is going to be my next purchase from him.

What did I get from his place this time, a PCW 8256 for $15! Couldn't believe it, a whole $15 for the monitor/main unit, keyboard, and printer. The problem is no software, so I have been unable to test it. my main interest was to take it apart and see how it ticks, but putting out this issue has prevented that task. So maybe by next time I will have a little inside information about these CP/M 3.0 machines.

Forth Day

Back in November of last year, the Forth Interest Group (FIG) had their yearly meeting in the San Francisco Bay area. This is pretty much a low key affair these days with a full day of speakers who talk about what they are working on. To start the program off, John Hall the president of FIG, started by reminding us that over 200,000 embedded Forth's are used everyday in Sun workstations.

A later speaker, Mitch Bradley had just returned from COMDEX where he talked on the new Open BOOT standard. This standard in essence is the same approach of using Forth to start and test interface cards as your system boots up. The players in this game however, are all current product vendors (like IBM and INTEL to start with). It is all part of the new PCI interface standards work, and you will be seeing and hearing more about this as the products start coming to market.

There were a number of companies showing their products and talking about how they were and still are able to under bid and develop products faster than others because they do it in Forth. My favorite product which has been making it to all the meetings over the years is the Talking Hand.

The talking hand is a long term project to develop a mechanical hand that can do finger spelling for deaf and blind people. It started originally at Stanford as a pet project of one of the electronic instructors and is about to be marketed. Over the years we have seen the hand and the computer equipment shrink from many STD bus cabinets to a single Z-World Z180 Forth based system.

Now the Z180 simply uses a look up table and uses commands received over the serial link from a PC to determine hand movements. So the text is processed by the PC program and could come from any source (keyboard, network, voice). The results however is a very small and much simpler controller and hand. I always enjoy seeing the hand and learning of it's progress.

The day ends with a talk by Chuck Moore who invented Forth. Chuck has recently been building his own CPU's. He developed his own special CAD program that designs and lays out the silicon structure. Chuck's latest CPU is called the MuP21. This 20Mhz CPU is very simple, uses Forth like instructions, simple stacks, and does calculations in amazing time.

Current testing put the machine at 80MIPS and Chuck felt that 150 MIPS would probably be it's limit. We watched one run and were amazed at how it is currently programmed. Chuck just sets bits for instructions he wants. It is all human programming here, no compliers needed or used (only 25 instructions). This is all part of Chuck's philosophy of making it very simple so you don't need expensive and complex tools. On the surface one might find his ideas hard to swallow, until you see what actually has been done.

My overall feeling was that Chuck is doing what used to be done in academia. Since so many colleges having funding problems, doing some sort of project that is 180 degrees out with the normal philosophy on the subject just doesn't happen anymore. Chuck is doing just that kind of work, and finding some buyers as well. So if you want to see what the very leading edge is all about contact Ting at Offete Press (415-574-8250) for Books and development systems.

Network Problems

Received this internet note that might help some of you would be users.

Sub: InterNIC Information Services

Bill,

Just received my copy of TCJ #64 and noticed Herb Johnson's question about Internet access.

May I suggest some information on InterNIC in the TCJ? InterNIC being a five year project sponsored by the National Science Foundation to provide services to the Internet community.

Their introductory fact sheets are a great help and the article "Internet Basics" by Roy Tennant is in the public domain. They can be reached at:

Telephone:	1-800-444-4345 1-619-455-4600
FAX:	1-619-455-3990
E-mail:	info@internic.net
Postal mail:	InterNIC Information Services PO Box 85608 San Diego, CA 92186-9784

Their information combined with the Internet information in TCJ #62(?) got me off to a flying start.

TKS for the TCJ,
Gary Welles
welles/117-8863@mcimail.com

TKS or Thanks too Gary for this advice. With that it is thanks for supporting TCJ and till next time....keep hacking.

Discover
The Z-Letter

The Z-letter is the only publication exclusively for CP/M and the Z-System. Eagle computers and Spellbinder support. Licensed CP/M distributor.

Subscriptions: $18 US, $22 Canada and Mexico, $36 Overseas. Write or call for free sample.

The Z-Letter
Lambda Software Publishing
149 West Hilliard Lane
Eugene, OR 97404-3057
(503) 688-3563

Advent Kaypro Upgrades

TurboROM. Allows flexible configuration of your entire system, read/write additional formats and more, only $35.

Replacement Floppy drives and Hard Drive Conversion Kits. Call or write for availability & pricing.

Call (916)483-0312
eves, weekends or write
Chuck Stafford
4000 Norris Ave.
Sacramento, CA 95821

TCJ MARKET PLACE

Advertising for small business

First Insertion:	$25
Reinsertion:	$20
Full Six issues	$100

Rates include typesetting. Payment must accompany order. VISA, MasterCard, Diner's Club, Carte Blanche accepted. Checks, money orders must be US funds. Resetting of ad consitutes a new advertisement at first time insertion rates.

Mail ad or contact
The Computer Journal
P.O. Box 535
Lincoln, CA 95648-0535

CP/M SOFTWARE

100 page Public Domain Catalog, $8.50 plus $1.50 shipping and handling. New Digital Research CP/M 2.2 manual, $19.95 plus $3.00 shipping and handling. Also, MS/PC-DOS Software. Disk Copying, including AMSTRAD. Send self addressed, stamped envelope for free Flyer, Catalog $1.00

Elliam Associates
Box 2664
Atascadero, CA 93423
805-466-8440

NEW MAGAZINE

the world of 68' micros

supporting
Tandy Color Computer
OS-9 & OSK

$23/year for 8 issues
$30/year Canada/Mexico
$35/year overseas

Published by:
FARNA Systems
P.O. Box 321
Warner Robins
GA 31099-0321

S-100/IEEE-696

IMSAI Altair
Compupro Morrow
Cromemco
and more!

Cards • Docs • Systems

Dr. S-100

Herb Johnson,
CN 5256 #105,
Princeton, NJ 08543
(609) 771-1503

THE FORTH SOURCE

Hardware & Software

MOUNTAIN VIEW PRESS

Glen B. Haydon, M.D.
Route 2 Box 429
La Honda, CA 94020

(415) 747 0760

PCB's in Minutes From LaserPrint!*

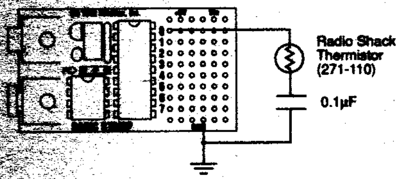
8 1/2" x 11"
Sheets
100% MBG

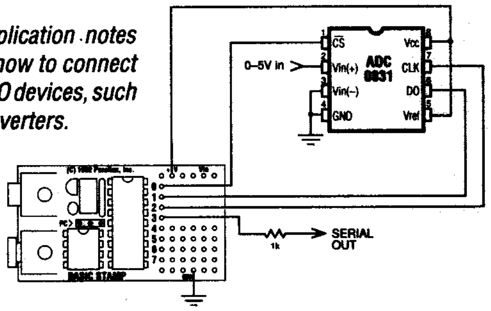
* Or Photocopier
Use household
iron to apply.

PnP BLUE or **PnP WET**

PnP BLUE	**PnP WET**
For High Precision Professional PCB Layouts	Easy Hobby Quality PCB's
1. LaserPrint 2. Iron-On 3. Peel-Off 4. Etch	1. LaserPrint 2. Iron-On 3. Soak-Off w/ Water 4. Etch
An Extra Layer of Resist for Super Fine Traces	Transfers Laser or Copier Toner as Resist

20Sh$30/40Sh$50/100Sh$100 Blue/Wet (No Mix)
Sample Pack 5 Shts Blue + 5 Shts Wet $20
VISA/MC/PO/CK/MO $4 S&H -- 2nd Day Mail
Techniks Inc. P.O. Box 463 Ringoes NJ 08551
(908)788-8249





TCJ The Computer Journal
Post Office Box 535
Lincoln, CA 95648-0535
United States

Telephone: (916) 645-1670